

1 2 9 0



UNIVERSIDADE D
COIMBRA

Adriano Del Pino Lino

**LABDER - LABORATÓRIO VIRTUAL DE ENSINO-
APRENDIZAGEM DE BANCO DE DADOS
RELACIONAIS: UMA ABORDAGEM DE AVALIAÇÃO
AUTOMÁTICA DE DIAGRAMAS ER E SQL**

**Tese no âmbito do Programa de Doutoramento em Ciências e Tecnologias da
Informação, orientada pelo Professor Doutor Álvaro Manuel Reis da Rocha e
Professor Doutor Luís Miguel Machado Lopes Macedo e apresentada ao
Departamento de Engenharia Informática da Faculdade de Ciências e
Tecnologia da Universidade de Coimbra.**

Maio de 2021

Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

**LABDER - LABORATÓRIO VIRTUAL DE ENSINO-
APRENDIZAGEM DE BANCO DE DADOS
RELACIONAIS: UMA ABORDAGEM DE AVALIAÇÃO
AUTOMÁTICA DE DIAGRAMAS ER E SQL**

Adriano Del Pino Lino

Tese no âmbito do Programa de Doutoramento em Ciências e Tecnologias da Informação, orientada pelo Professor Álvaro Manuel Reis da Rocha e Professor Doutor Luís Miguel Machado Lopes Macedo, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Maio de 2021



UNIVERSIDADE D
COIMBRA

Patrocinadores



UNIVERSIDADE FEDERAL DO OESTE DO PARÁ



Parceiros



UNIVERSIDADE DE
COIMBRA



UNIVERSIDAD
NACIONAL
DE COLOMBIA



iscac.pt



Universidade Federal
da Grande Dourados



O trabalho de pesquisa apresentado nesta tese foi parcialmente realizado no escopo do projeto: Avaliação Automática do Ensino e Aprendizagem financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) número 202276/2015-5, e também parcialmente financiada pela licença de aptidão cedida pela Universidade Federal do Oeste do Pará (UFOPA) número 2.970/14, no âmbito do Plano Nacional de Formação de Pessoal, sob a supervisão do Ministério de Educação brasileiro. O apoio parcial para participar em eventos científicos foi fornecido pela Associação Ibérica de Sistemas e Tecnologias de Informação (AISTI) e pelo Centro de Informática e Sistemas da Universidade de Coimbra (CISUC). Este projeto de pesquisa também teve o apoio parcial da empresa OctoPerf, que cedeu as licenças e o suporte técnico necessário para realizar o teste do LabDER.

Este trabalho de pesquisa teve parceiras com as seguintes instituições de ensino: Universidade de Coimbra (UC), Portugal; Universidade Pública da Colômbia (UNAL), Colômbia; National Institute of Technology Raipur (NIT), Índia; Instituto Politécnico do Cávado e do Ave (IPCA), Portugal; Coimbra Business School (ISCAC), Portugal; Universidade Federal de Grande Dourados (UFGD), Brasil; e Faculdade Pan Amazônica (FAPAN), Brasil.

Dedicado à minha esposa Amanda e filhas Alice e Alexia

AGRADECIMENTOS

Agradeço a Deus e a Nossa Senhora, por me conceder o dom da vida, pela saúde e união da minha família, permitindo que desfrute, em plenitude, os prazeres da alma.

Aos meus pais, Maria Tereza e Mozart Lino que, desde sempre, me incentivaram a estudar e a acreditar que é possível conquistar os nossos objetivos, através de muito trabalho e estudo.

Um agradecimento especial à minha amada esposa Amanda, que com paciência cuidou das nossas filhas Alice e Alexia durante as minhas ausências, mantendo a família sempre unida. Como se não bastasse, também me ajudou em diferentes fases da tese, com o seu conhecimento em engenharia de software. Além disso, teve sempre o bom discernimento de me manter sereno, naqueles momentos em que eu mais precisava. Também me ajudou sempre com as tarefas domésticas, mimando-me sempre muito. Cada artigo publicado foi sempre celebrado com muita cumplicidade e alegria. Obrigado por me permitires gozar a vida, ao longo de tantos momentos agradáveis, ao lado de nossas filhas, e por me ensinares a ser um bom pai e marido.

Tenho de agradecer a orientação do Professor Doutor Álvaro Manuel Reis da Rocha, que criou um perfeito ambiente de partilha de conhecimento, incentivando, com humor, a conclusão desta tese com a frase “chute a bola para a frente”. Os seus conhecimentos sobre padrões internacionais de qualidade de software foram vitais para criar um sistema robusto, passível de ser utilizado internacionalmente. Além do aspeto da investigação, ainda tive o seu apoio financeiro para participar em eventos científicos, e o suporte emocional para manter a minha integridade física e mental.

Igualmente agradeço ao co-orientador Professor Doutor Luís Miguel Machado Lopes Macedo que, durante as suas aulas, contribuiu com a perspectiva de aprendizagem computacional para abordar o tema desta tese. Durante as nossas reuniões, esclareci diversas dúvidas técnicas do campo, e com sua vasta maturidade, sempre recebi os alertas dos caminhos a serem evitados. Recebi o conhecimento necessário de ambos os orientadores, que também abriram portas para a colaboração internacional, ampliando o horizonte da aplicação desse projeto permitindo, assim, o meu desenvolvimento como investigador.

Deixo também uma palavra de apreço aos oito professores que me confiaram cerca de trezentos alunos e que participaram na experimentação do sistema desenvolvido. O rigor e a permanente disponibilidade de todos os professores proporcionaram a efetivação do plano

de trabalho e garantiram a efetiva participação dos alunos na recolha de dados. De forma particular, agradeço aos professores, Isabel Pedrosa, Nestor Dário, Olga Coutinho, Paulo Adriano, Maxwell Dias, Pavan Mishra, e Mónica Guayasamin, por toda a dedicação.

Estendo os agradecimentos aos funcionários do DEI/UC e aos colegas de laboratório: Ana, Ali, Adilson, André, Diogo, Jorge, Paulo, Rahul, Ronaldo, João, Phong, e Omar. Grato pela prontidão e partilha de conhecimento, sempre seguido de um cafezinho.

Estou muito grato por ter recebido tanto amor e palavras de incentivo de todos os meus amigos e parentes. Amigos como a Yana e o João Alvo que abriram as portas de suas casas, trazendo alegria à minha família. Foi muito divertido e gratificante ver as nossas filhas brincando. Fiz muitas amizades, como com o Omar, o Rahul e a Andreia, que me falaram das suas culturas, respetivamente, da Síria, Índia e Portugal. Essa riqueza cultural permitiu celebrar, até duas vezes, o Natal, quer seguindo a tradições Russa quer a Portuguesa. Até aprendi a vindimar, isto com o João Veríssimo e a família dele, que esteve sempre presente, dando-me o seu constante apoio. Devo também agradecer àqueles parentes, como o Euclides Filho, que sempre me proporcionaram harmonia familiar, quando necessário. Ou então, àqueles parentes como os meus sogros, a quem estou extremamente grato pelo zelo e dedicação para com as minhas filhas, permitindo dedicar-me, exclusivamente, aos estudos com a minha esposa.

Faço uma menção especial ao Professor Doutor Peter Pin-Shan Chen, e ao diretor executivo do Instituto de Engenharia de Software da Universidade Carnegie Mellon, Doutor Paul Nielsen, que se disponibilizaram para trocar experiências no evento ER 2019. Foi muito especial poder confirmar o levantamento histórico sobre a modelagem de base de dados, descrita nesta tese, justamente com as pessoas que as desenvolveram. Os elogios e feedbacks positivos acerca da investigação desenvolvida não foram, senão, um dos pontos mais importantes deste trabalho.

Agradeço ainda: a oportunidade do financiamento desta investigação pelo CNPq/CAPES – Programa Ciência sem Fronteiras – CsF (202276/2015-5); bem como a licença de aptidão cedida pela Universidade Federal do Oeste do Pará (2.970/14); o apoio, para poder participar em eventos científicos, dado pela AISTI - Associação Ibérica de Sistemas e Tecnologias de Informação; ao CISUC - Centre for Informatics and Systems, centro de investigação que me acolheu; e a parceria com a empresa de teste de software OctoPerf, que cedeu as licenças e o suporte técnico necessário.

RESUMO

A procura de profissionais de informática da área de Bases de Dados tem crescido recentemente e superado as projeções mais otimistas. Esse aumento pressiona as instituições de ensino a oferecer especialistas no domínio. Neste contexto, surgem cursos de Base de Dados com muitos alunos que requerem demasiados recursos humanos para acompanhar as atividades práticas de laboratório, sendo os ambientes virtuais de avaliação automática um forte aliado na resolução desse cenário. Entretanto, são raras as instituições que possuem ambientes virtuais para o ensino de bases de dados que automaticamente avaliem as respostas dos alunos, especificamente respostas diagramáticas. As abordagens existentes de avaliação automática possuem uma série de problemas, que resultam em sistemas que mostram apenas um *feedback* correto ou incorreto, e não permitem aos alunos aprimorarem a solução.

O trabalho descrito nesta tese investigou como avaliar automaticamente diagramas de entidade e relacionamento que pudessem ser aplicados no contexto das disciplinas introdutórias de Bases de Dados em cursos de licenciatura em informática. O projeto seguiu a metodologia de investigação *Design-Science Research*, onde primeiramente foram identificados os problemas das abordagens de avaliação automática, como (a) interface restritiva, (b) resposta restritiva, (c) destaques visuais, (d) notação restritiva, (e) avaliação sem equivalência, (f) *feedback* inadequado, e (g) *feedback* estático. O método seguiu um processo iterativo de definição dos objetivos, métodos e procedimentos que levaram a propor soluções para automatizar a avaliação de diagramas no contexto de ensino e aprendizagem de bases de dados relacionais. Como resultado dessa investigação foi obtido um ambiente virtual de ensino de bases de dados relacionais, denominado LabDER, que avalia automaticamente as respostas dos alunos.

O LabDER automatiza a avaliação de diagramas de entidade e relacionamento, SQL, questões de escolha múltipla e de texto livre. Diferente de outras abordagens, o LabDER fornece um *feedback* semântico com a nota final do aluno baseado em técnicas de compilador, métricas de engenharia de software e aprendizagem computacional. Essa abordagem inspira o aluno a procurar a melhor resposta, ou seja, ajuda-o a encontrar a resposta estimada ou superior à solução do sistema.

O ambiente LabDER passou por um minucioso processo de verificação e validação através de testes de funcionalidade, desempenho e usabilidade, e foi usado satisfatoriamente

por 322 alunos e 8 professores de universidades do Brasil, Colômbia, Portugal e Índia. Cerca de 98% dos utilizadores aceitaram a usabilidade do sistema e 95% avaliaram o sistema como melhor imaginável na escala *System Usability Scale*. Garante também uma avaliação automática com tempo de resposta médio de menos de 1 segundo para 500 utilizadores em simultâneo.

Foi recolhida uma base de dados com mais de 40 mil respostas, das quais 5 mil foram tratadas para ter dados suficientes de treino de modelos preditivos de diagramas de entidade e relacionamento. As questões expostas aos alunos foram cuidadosamente construídas, seguindo o plano de ensino do professor e distribuídas ao longo do semestre letivo em seis universidades com diferentes níveis de complexidade. O modelo preditivo com uma *precisão* acima de 90% foi obtido seguindo a metodologia CRISP-DM, após uma análise de 2.700 modelos.

A abordagem de avaliação automática do LabDER oferece as seguintes vantagens: (a) aceita várias respostas corretas por questão; (b) prevê as notas finais de cada resposta do aluno, baseado no modelo preditivo que mede a distância entre a resposta do aluno e a resposta do professor pelas métricas, (c) ajuda a identificar o erro da solução submetida pelo *feedback* que associa erros a conceitos de BD; (d) oferece ao aluno a opção de refazer a solução; (e) oferece ao professor um sistema que pode ser usado em avaliações diagnósticas, formativas e sumativas. No futuro, espera-se que esta abordagem possa ser adaptada para avaliar outros tipos de diagramas e linguagens de programação.

Palavras-chave: bases de dados, diagramas de entidade e relacionamento, avaliação automática, ambiente virtual, métricas, compiladores, aprendizagem computacional.

ABSTRACT

Recently the demand for computing experts on Database systems has been growing and exceeding the highest expectations. This rise has imposed on the educational system and institutions the supply of experts in this field. In this scenario, introductory database subjects are faced with too many students who require a large human resources capacity to manage and monitor the practical laboratory activities. So virtual environments of automatic assessment are a strong ally to solve this scenario. Nonetheless just a few of them are able to offer a virtual learning environment for Database teaching in order to automatically assess student's answers, explicitly diagrammatic answer. The current approaches to automatic assessment are actually coming up with a few problems which just show a correct or incorrect feedback, not allowing students to improve on any answer.

The research made on this thesis was aimed at finding a way to automatically assess entity relationship diagrams. Further this research aimed to suit Database systems subjects in higher education degrees on Computer Science. This project followed the Design-Science Research Methodology. At first, problems with automatic assessment approaches were noticed, such as (a) restrictive interface, (b) restrictive answer, (c) visual highlights, (d) restrictive notation, (e) assessment with no equivalence, (f) unsuitable feedback, and (g) static feedback. The method followed an interactive process for several goal definitions, methods and procedures. This lead to the proposal of several solutions in order to automatize the diagram assessment in the learning & teaching context of Database systems. This research resulted in a virtual learning environment for teaching Database systems – known as LabDER – which automatically assesses students' answers.

LabDER automates the assessment of entity-relationship diagram, SQL, multiple choice and subject questions responses. Differing from other approaches, LabDER provides a semantic feedback with the student's grade based using a techniques blend of compiler, software engineering metrics and machine learning. This method motivates students to achieve the best possible answer, i.e., it helps them to find the expected answer or even a superior answer than the system's solution.

The LabDER system has been run in a precise and accurate software testing, based on functionality, performance and usability tests. It was satisfactorily used by three hundred and twenty-two students and eight professors from different universities in Brazil, Colombia, Portugal and India. About 98% of the users have classified the system as acceptable and 95%

evaluated the system as the best ever possible on the System Usability Scale. Automatic assessment also guarantees an average answer timing within less than one second for up to five hundred simultaneous users.

A dataset of over forty thousand answers was collected. Five thousand of them were handled in order to get enough training data to determine patterns in entity relationship diagram. Along the semester, questions with different complexity levels were carefully presented to the students following the professors teaching plan in the six universities. After performing the modeling process for two thousand and seven hundred times, following CRISP-DM methodology, it was possible to get results with a precision close to 90%.

The following advantages are provided by LabDER automatic assessment approach: (a) it accepts several correct answers to each question; (b) it provides a grade for each student's answer, based on the prediction model which measures the distance between student's answer and the professor's answer by the metrics; (c) it helps to identify the error in the submitted solution with the feedback that associates the errors with DB concepts; (d) it gives the student the option to redo the solution; (e) it gives the professor a system that may be used either in diagnostic, formative or summative assessments. In the future, it is expected that this approach can be adopted to evaluate different diagrammatic representations and programming languages.

Keywords: database, entity-relationship diagram, automatic assessment, virtual learning environment, metrics, compilers, machine learning.

ÍNDICE

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	6
1.3	Metodologia de Investigação	8
1.4	Contributos	11
1.5	Estrutura da tese	12
2	Fundamentação teórica	13
2.1	Modelo ER	13
2.2	Diagramas de Entidade e Relacionamento	15
2.2.1	Notações dos diagramas de entidade e relacionamento	19
2.2.2	Métodos de Avaliação de DER	25
2.2.3	Desafios na Avaliação do DER	30
3	Ferramentas de Avaliação Automática de DER	41
3.1	Considerações iniciais	41
3.2	Abordagens de avaliação de DER	42
3.2.1	Sistema tutores inteligentes	42
3.2.2	Processamento de linguagem natural	43
3.3	Avaliação dos ambientes	43
3.3.1	OpenMark	44
3.3.2	EER-Tutor	46
3.3.3	ERM-VLE	48
3.3.4	DATSys	51
3.3.5	Principais problemas	52
4	Framework de Avaliação Automática de DER e SQL	57
4.1	Considerações iniciais	57
4.2	Breve histórico	57
4.3	Abordagem da avaliação automática de DER	61
4.3.1	Primeiro nível de avaliação - compiladores	65
4.3.2	Segundo nível de avaliação - métricas	74
4.3.3	Terceiro nível de avaliação - aprendizagem computacional	84

5	Ambiente de Ensino e Aprendizagem LabDER	87
5.1	Processo de Desenvolvimento do LabDER	87
5.1.1	Análise de Requisitos	90
5.1.2	Projeto	95
5.1.3	Desenvolvimento	101
5.2	Teste do LabDER	112
5.2.1	Tipos de Teste do LabDER	112
5.2.2	Processo de teste	113
5.3	Teste de Funcionalidade	115
5.3.1	Plano	116
5.3.2	Projeto	119
5.3.3	Execução e resultados	121
5.4	Teste de Desempenho	122
5.4.1	Definições	122
5.4.2	Contexto e objetivos	124
5.4.3	Processo de Teste de Desempenho	127
5.4.4	Plano	130
5.4.5	Projeto	132
5.4.6	Execução	137
5.4.7	Análise de resultados	139
5.5	Teste de Usabilidade	148
5.5.1	Questionário SUS	149
5.5.2	Plano	152
5.5.3	Projeto	153
5.5.4	Execução e Resultado	155
6	Implantação e Operação	158
6.1	Considerações iniciais	158
6.2	Preparação do ambiente	158
6.3	Processo de conceção do modelo	168
6.3.1	Primeiro ciclo	169
6.3.2	Segundo ciclo	186
6.3.3	Terceiro ciclo	199
6.4	Avaliação de campo do LabDER	207
6.4.1	População	207
6.4.2	Metodologia e instrumentos	209
6.4.3	Análise de resultados	211
7	Conclusões	225

7.1	Considerações iniciais	225
7.2	Contribuições e reflexões	226
7.3	Limitações da investigação	228
7.4	Trabalhos futuros	229
	Publicações resultantes deste trabalho de investigação	232
	Referências Bibliográficas	233
	Anexo A: Lista de livros de BD	249
	Anexo B: Teste de Desempenho	250
	Anexo C: Caso de Teste - Perfil Aluno	296
	Anexo D: Caso de Teste - Perfil Professor	303
	Anexo E: Plano de atividades para uso do LabDER	313
	Anexo F: Métricas - Análises de Correlação - 1º Ciclo	318
	Anexo G: Métricas - Análises de Correlação - 2º Ciclo	319
	Anexo H: Resultado dos ensaios - 1º Ciclo	320
	Anexo I: Resultado dos ensaios - 2º Ciclo	322
	Anexo J: Resultado dos ensaios - 3º Ciclo	324
	Anexo K: Configurações para a conceção do modelo de aprendizagem	326

LISTA DE FIGURAS

Figura 1-1. Método DSRM de investigação da tese.	9
Figura 2-1. Comparação das notações de modelo ER, adaptado de (Coronel & Morris, 2019).	19
Figura 2-2. Exemplo de diagrama na notação CHEN, extraído de (Oppel, 2011).	21
Figura 2-3. Exemplo de diagrama na notação IDEF1X, extraído de (Oppel, 2011).	21
Figura 2-4. Exemplo do DER na notação engenharia de informação, extraída da revista nº 5 publicada na Computerworld (Finkelstein, 1981b).	23
Figura 2-5. Exemplo de diagrama na notação <i>crow's foot</i> , extraído de (Oppel, 2011).	24
Figura 2-6. Quantidade de livros na área de BD com tópico de DER com exercícios e material de suporte para o professor (“Sim” em verde - livros selecionados, e “Não” em vermelho - livros rejeitados).	33
Figura 2-7. Exemplo de DER equivalentes para guardar o grau de escolaridade dos funcionários.	36
Figura 2-8. Número de possibilidades de relacionamentos por entidades.	39
Figura 3-1. Visão do aluno para ler uma questão no Moodle e aceder a ferramenta OpenMark (Thomas, 2013).	44
Figura 3-2. A ferramenta OpenMark na visão do aluno (Thomas, 2013).	45
Figura 3-3. Interface ERR-Tutor.	47
Figura 3-4. Interface VLE-ERM.	50
Figura 3-5. Interface do DATsys com a resposta DER do aluno.	52
Figura 4-1. Arquitetura do LabSQL, extraído de (Melo et al., 2013).	58
Figura 4-2. Uma comparação entre as arquiteturas do LabSQL (a), um compilador (b), e um simulador de respostas para análise de erros (c).	60
Figura 4-3. <i>Framework</i> de Avaliação Automática de DER e SQL.	61
Figura 4-4. Processo de funcionamento de um compilador (Singh et al., 2009)	62
Figura 4-5. Exemplo de DER com uma tabela e duas colunas com código SQL correspondente.	66
Figura 4-6. Fluxo do programa principal do analisador léxico.	68
Figura 4-7. Exemplo de erro - Identificador ambíguo.	68
Figura 4-8. Exemplo de erro - Identificadores repetidos.	69
Figura 4-9. Exemplo de tabela de símbolo para verificação de tipo de dados do DER.	72
Figura 4-10. Exemplo de um diagrama de entidade e relacionamento (Coronel & Morris, 2019).	78
Figura 4-11. Dois exemplos de diagramas ER.	80
Figura 4-12. Entrada e saída da fase aprendizagem computacional - DER.	85
Figura 5-1. Modelo de desenvolvimento cascata, adaptado de (Sommerville, 2016).	90
Figura 5-2. Arquitetura cliente-servidor inicial do ambiente LabDER.	96
Figura 5-3. Visão geral da arquitetura de software do LabDER.	97
Figura 5-4. Editor de diagrama de entidade e relacionamento.	100
Figura 5-5. Interface de login do LabDER.	102
Figura 5-6. Painel principal - Perfil do professor.	103
Figura 5-7. Relatório analisar resultados - Perfil do professor.	105
Figura 5-8. Janela de execução de exercício SQL - Perfil do professor.	106
Figura 5-9. Janela de execução de exercício DER - Perfil do professor.	107
Figura 5-10. Janela menu principal - Perfil do aluno.	108
Figura 5-11. Janela para visualizar o livro digital - Perfil do aluno.	109
Figura 5-12. Janela de execução de exercício SQL - Perfil do aluno.	110
Figura 5-13. Janela de execução de exercício DER - Perfil do aluno.	111

Figura 5-14. Fluxo parcial do processo padrão IEEE 829-2008. Figura adaptada de (IEEE Computer Society, 2008).	114
Figura 5-15. Gráfico de proporção de problemas encontrados por número de testadores. Adaptado de (Nielsen & Landauer, 1993).	119
Figura 5-16. Fluxo de execução do caso de teste de funcionalidade.	120
Figura 5-17. Fluxo do processo de teste de desempenho.	128
Figura 5-18. Casos de testes implementados na ferramenta Octoperf.	135
Figura 5-19. Configuração do TC - Aluno DER no Octoperf.	136
Figura 5-20. Fluxo de testes e intervenções em ambientes web.	137
Figura 5-21. Versão final da arquitetura do ambiente LabDER.	138
Figura 5-22. Tempo de resposta médio do TC - Aluno DER por container com 1vCPU e 2vCPU.	140
Figura 5-23. Tempo de resposta médio do TC - Aluno DER por requisições com 1vCPU e 2vCPU. ..	141
Figura 5-24. Tempo de resposta médio do TC - Aluno Visualizar Conteúdo por container 1vCPU e 2vCPU.	143
Figura 5-25. Número de erros do TC - Aluno Visualizar Conteúdo com 1vCPU e 2vCPU.	144
Figura 5-26. Tempo de resposta médio do cenário 4 do TC - Professor Analisa Resultado por requisições com 1vCPU e 2vCPU.	146
Figura 5-27. Comparação de pontuações médias por quartil, classificação dos adjetivos e aceitabilidade da pontuação geral do SUS, extraída de (Bangor et al., 2008).	151
Figura 5-28. Pontuações de usabilidade para os 14 produtos mapeados nas faixas de aceitabilidade propostas por Bangor (2009). Nota. GPS = sistema de posicionamento global; DVR = gravador de vídeo digital; PPT = PowerPoint; ATM = caixa eletrônico.	151
Figura 5-29. Implementação do formulário SUS no Google Forms e LabDER	154
Figura 5-30. Acompanhamento da avaliação FAPAN.	155
Figura 5-31. Classificação de usabilidade do LabDER (72,97) na escala de aceitabilidade (Bangor et al., 2009). As pontuações abaixo de 50 são consideradas inaceitáveis.	155
Figura 5-32. Distribuição de frequência de usabilidade do LabDER para classificação de adjetivos do SUS.	156
Figura 5-33. Classificações de usabilidade do LabDER em percentil, com 64 utilizadores.	156
Figura 6-1. Controle dos dados trocados entre professores e investigador via DropBox.	160
Figura 6-2. Cronograma de atividades de preparação e execução do LabDER no Project.	161
Figura 6-3. Exemplo de mapeamento do conteúdo programático do professor e o material do LabDER.	162
Figura 6-4. Utilizadores ativos na primeira avaliação formal da UNAL.	166
Figura 6-5. Total de submissões no LabDER por tipo de questão.	166
Figura 6-6. Gráfico da frequência de submissões por tipo de questão e universidade.	167
Figura 6-7. Modelo CRISP-DM (Chapman et al., 2000).	168
Figura 6-8. Atividades do CRISP-DM, extraído de (Chapman et al., 2000).	169
Figura 6-9. Dados brutos da tabela de métricas.	173
Figura 6-10. Importância das variáveis de entrada no 1º ciclo.	176
Figura 6-11. Frequência dos atributos no 1º ciclo.	177
Figura 6-12. Frequência das cinco categorias de avaliação no 1º Ciclo.	178
Figura 6-13. Proporção de dados de formação e teste em nove ensaios.	180
Figura 6-14. Representação gráfica do procedimento de validação cruzada de dez vezes.	180
Figura 6-15. Abordagem experimental de nove ensaios de modelos preditivos.	181

Figura 6-16. Gráfico de comparação de desempenho (<i>precision</i> , <i>recall</i> , <i>F1-Score</i> , e <i>accuracy</i>) por algoritmo de classificação (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD e SVC) com os dados do 1º ciclo.	182
Figura 6-17. Gráfico de <i>precision</i> para os dados do primeiro ciclo.	183
Figura 6-18. Desempenho do <i>recall</i> para os dados do primeiro ciclo.	184
Figura 6-19. Desempenho do <i>F1-Score</i> para os dados do primeiro ciclo.	184
Figura 6-20. Desempenho de <i>accuracy</i> para os dados do primeiro ciclo.	185
Figura 6-21. Captura das medidas de desempenho AUC e matriz de confusão de forma aleatória no 1ª ciclo.	187
Figura 6-22. Exemplo de classificações diferentes para respostas de DER com mesmo número de componentes de DER, lado esquerdo 3 e lado direito 4.	189
Figura 6-23. Resposta do professor.	190
Figura 6-24. Correlação dos atributos de entrada no 2º ciclo.	191
Figura 6-25. Comparação entre a correlação dos atributos no 1º e 2º ciclo.	192
Figura 6-26. Frequência dos atributos no 2º ciclo.	192
Figura 6-27. Frequência das cinco categorias de avaliação no 2º Ciclo.	193
Figura 6-28. Gráfico de comparação de desempenho (<i>precision</i> , <i>recall</i> , <i>F1-Score</i> , e <i>accuracy</i>) por algoritmo de classificação (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD e SVC) com os dados do 2º ciclo.	194
Figura 6-29. Gráfico comparativo de desempenho entre os 1ª e 2ª ciclos.	196
Figura 6-30. Desempenho de <i>precision</i> para os dados do segundo ciclo.	197
Figura 6-31. Desempenho do <i>recall</i> para os dados do segundo ciclo.	198
Figura 6-32. Desempenho do <i>F1-Score</i> para os dados do segundo ciclo.	198
Figura 6-33. Desempenho de <i>accuracy</i> para os dados do segundo ciclo.	199
Figura 6-34. Gráfico de comparação de desempenho (<i>precision</i> , <i>recall</i> , <i>F1-Score</i> , e <i>accuracy</i>) por algoritmo de classificação (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD e SVC) com os dados 3º ciclo.	201
Figura 6-35. Gráfico comparativo de desempenho entre os 1º, 2º, e 3º ciclos.	203
Figura 6-36. Desempenho de <i>precision</i> para os dados do 3º ciclo.	204
Figura 6-37. Desempenho do <i>recall</i> para os dados do 3º ciclo.	205
Figura 6-38. Desempenho do <i>F1-Score</i> para os dados do 3º ciclo.	205
Figura 6-39. Desempenho de <i>accuracy</i> para os dados do 3º ciclo.	206
Figura 6-40. ISCAC - Área comum de estudo com acesso à internet.	208
Figura 6-41. Última avaliação formal aplicada no ISCAC, a 11 de julho de 2019.	209
Figura 6-42. Utilizadores do ISCAC em Portugal durante os meses de fevereiro a julho de 2019.	213
Figura 6-43. Interface do professor que avalia o DER do aluno com <i>feedback</i> automática e manual - Tipo de questão Transcrição.	216
Figura 6-44. Interface do professor que avalia o DER do aluno com <i>feedback</i> automático - Tipo de questão Cenário.	218
Figura 6-45. Interface do professor que avalia o SQL da resposta do aluno.	219
Figura 6-46. Interface professor que avalia a resposta de escolha múltipla do aluno.	220
Figura 6-47. Interface do professor que avalia a resposta de texto livre do aluno.	221

LISTA DE TABELAS

Tabela 2-1. Níveis de abstração do dado, baseado em (Coronel & Morris, 2019).	16
Tabela 2-2. Principais conceitos de BD usados no DER, extraído de (Coronel & Morris, 2019).	17
Tabela 2-3. Níveis de modelo por conceitos de BD, uma adaptação dos níveis de estilo de construções do IDEF1X (ISO/IEC/IEEE, 2012).	18
Tabela 2-4. Dimensões de qualidade de dados padrão ISO/IEC 25012:2008, extraído de (Batini & Scannapieco, 2016).	26
Tabela 2-5. Uma estrutura de classificação para dimensões de qualidade de dados e informações, extraído de (Batini & Scannapieco, 2016).	27
Tabela 2-6. Tipos genéricos de questões de DER.	34
Tabela 3-1. Características das ferramentas de avaliação de DER.	56
Tabela 4-1. Métricas para medir o diagrama de entidade e relacionamento no LabDER.	77
Tabela 4-2. Valores métricos para o diagrama ER da figura anterior.	79
Tabela 4-3. Abstrações de medida para as métricas de complexidade estrutural do DER.	83
Tabela 4-4. Rótulos linguísticos de compreensão.	86
Tabela 5-1. Lista de requisitos funcionais do perfil do professor.	94
Tabela 5-2. Lista de requisitos funcionais do perfil do aluno.	95
Tabela 5-3. Descrição dos requisitos não funcionais (RNF).	95
Tabela 5-4. Comparação das APIs para edição de DER.	99
Tabela 5-5. Tópicos abordados no livro digital do LabDER.	109
Tabela 5-6. Definição do plano de testes software do LabDER.	113
Tabela 5-7. Lista de Casos de Teste (TC) por Requisitos Funcionais (RF) do perfil do professor.	117
Tabela 5-8. Lista de Casos de Teste (TC) por Requisitos Funcionais (RF) do perfil do aluno.	118
Tabela 5-9. Formato do formulário do caso de teste de funcionalidade.	120
Tabela 5-10. Caso de teste 45 do Relatório - Analisar Resultados, perfil do professor.	121
Tabela 5-11. Objetivos dos testes de desempenho e desafios, adaptado de (Meier et al., 2007b).	126
Tabela 5-12. Desempenho de páginas web, extraído de (Rempel & Garret, 2015).	131
Tabela 5-13. Caso de teste (TC) de desempenho por perfil do utilizador de sistema e cenários de utilizadores virtuais (UV).	132
Tabela 5-14. Critérios aceitáveis de tempo de resposta por caso de teste do LabDER.	133
Tabela 5-15. Comparação de desempenho do cenário 4 do TC - Aluno DER.	139
Tabela 5-16. Comparação de desempenho do cenário 4 do TC - Aluno Visualizar Conteúdo.	142
Tabela 5-17. Comparação de desempenho do cenário 4 do TC - Aluno SQL.	145
Tabela 5-18. Comparação de desempenho do cenário 4 do TC - Professor Analisa Resultado.	145
Tabela 5-19. Comparação de desempenho do cenário 4 do TC - Professor Importar CSV.	147
Tabela 5-20. Questionário padrão do <i>System Usability Scale</i> (SUS).	150
Tabela 5-21. Total de participantes do questionário SUS.	153
Tabela 6-1. Cronograma genérico de atividades do projeto.	159
Tabela 6-2. Universidades que efetivamente utilizaram o módulo de DER.	160
Tabela 6-3. Quadro com resumo da compatibilidade dos programas do professor e LabDER.	163
Tabela 6-4. Resumo das atividades a serem executadas no LabDER.	164
Tabela 6-5. Quantidade de utilizadores por curso que tiveram atividades DER.	165
Tabela 6-6. Comparação do desempenho dos algoritmos com os dados do 1º ciclo.	183
Tabela 6-7. Critério de avaliação do DER para novas avaliações de DER.	188
Tabela 6-8. Comparação do desempenho dos algoritmos com os dados do 1º e 2º ciclo.	195

Tabela 6-9. Correlação dos novos atributos versus atributos originais.	200
Tabela 6-10. Comparação do desempenho dos algoritmos com os dados do 1º, 2º e 3º ciclo.	202
Tabela 6-11. Plano de atividades do curso por período e conteúdo. SQL é denominado como (S), DER é denominado (E), questões de texto livres são denominadas (D) e questões de escolha múltipla são denominadas (M).	210
Tabela 6-12. Número de respostas enviadas por tipo de questão por log geral do sistema.	211
Tabela 6-13. Frequência e percentagem de acerto dos componentes do DER avaliado no ISCAC, por métricas e atividades.	214
Tabela 6-14. Questionário de avaliação do LabDER aplicado aos professores.	222

LISTA DE ABREVIATURAS

BD	Bases de Dados
CASE	<i>Computer-aided software engineering</i>
CRISP-DM	<i>CRoss Industry Standard Process for Data Mining</i>
DER	Diagrama de Entidade e Relacionamento
EER	Entidade e Relacionamento Estendido
ER	Entidade e Relacionamento
IDEF1X	<i>Integration DEFinition for information modeling</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISO/IEC	<i>International Organization for Standardization and International Electrotechnical Commission</i>
MER	Modelo de Entidade e Relacionamento
PLN	Processamento de Linguagem Natural
SGBD	Sistemas de Gestão de Base de Dados
STI	Sistema Tutor Inteligente
SUS	<i>System Usability Scale</i>
UML	<i>Unified Modeling Language</i>

1.1 Enquadramento

É crescente a procura por sistemas cada vez mais integrados que ofereçam uma diversidade de informações mais precisas aos utilizadores. Um sistema na área da saúde que visa auxiliar na distribuição de medicamentos poderia ser mais preciso caso estivesse integrado com outros sistemas que contenham dados dos históricos de vacinação de utentes, registo de nascimento, ou sistemas de cuidados de saúde primário, entre outros. Esses sistemas integrados tendem a ser mais complexos, não apenas por atacarem uma gama de problemas de diferentes domínios, por exemplo problemas da área da saúde, mobilidade, segurança e educação, mas pela difícil tarefa de entender como esses sistemas se podem relacionar e partilhar um conjunto de dados. Nesse sentido, a modelação de dados ajuda os profissionais de informática na tarefa de entendimento do negócio, já que visa descrever a semântica da aplicação num diferente nível de abstração. A modelação de dados ocupa o cerne no processo de engenharia de software, e, a partir de um processo formal, pode ser obtido um documento que sintetiza todo o funcionamento do sistema.

O mundo produz um volume de dados cada vez maior, anualmente. De acordo com o relatório da IDC (2018), o volume global de dados chegará a 175 Zettabytes em 2025 e os dados estarão dispersos entre os *data centers* tradicionais e em nuvem, nas redes de telecomunicações e filiais, e nos *PCs*, smartphones e dispositivos de Internet das Coisas (IdC) (Reinsel, Gantz, & Rydning, 2018). A CISCO aponta que, globalmente, o tráfego de dados móveis aumentará sete vezes entre 2017 e 2022. As projeções da CISCO sugerem que em 2022 a região da América do Norte terá 352 milhões de utilizadores internet (cerca de 94% da população) com uma velocidade média de 94Mbps, na Europa Ocidental 380 milhões de utilizadores (62% da população), e na América Latina cerca de 465 milhões de utilizadores com acesso à internet (69% da população) com uma velocidade de 28.1Mbps (Cisco VNI, 2018). Reinsel et al. (2018) disse que “*A responsabilidade de manter e gerir*

todos esses dados de clientes e negócios apoia o crescimento nos data-centers de provedores na nuvem. Como resultado, o papel da empresa como administrador de dados continua a crescer e os consumidores não estão apenas permitindo isso, mas à espera disso. A partir de 2019, mais dados serão armazenados no núcleo da empresa do que em todos os terminais existentes no mundo” texto traduzido de (Reinsel et al., 2018).

Na decorrência dessa imensa massa de dados, as oportunidades para profissionais da área de bases de dados têm sido maiores do que as previsões mais otimistas. Segundo investigadores da IBM (2017), anteriormente nos Estados Unidos foi estimado 2.8 milhões de empregos na área de ciências e análise dos dados para o ano de 2018, no entanto esse número foi aproximadamente alcançado no ano de 2015 (Markow, Braganza, Taska, Miller, & Hughes, 2017). O relatório sobre os futuros empregos da *World Economic Forum* (2018) também menciona a procura emergente de profissionais da área de Ciências e Análise dos Dados, ao redor do mundo, para diversos setores da indústria como o setor automóvel, aeroespacial, transporte, química, materiais avançados e biotecnologia, saúde e cuidados de saúde globais, e indústrias de tecnologias de informação e comunicação (SCHWAB, 2018).

À medida que cresce a procura por trabalhadores de Ciências e Análise dos Dados, esse crescimento pressiona as instituições de ensino a oferecer especialistas no domínio. Entretanto, existe uma preocupação crescente de que a oferta de trabalhadores da área de Bases de Dados (BD) esteja muito deficitária em relação à procura. O programa de investigação e desenvolvimento de redes e tecnologia da informação dos Estados Unidos (NITRD), por exemplo, argumenta que devemos “*melhorar o cenário nacional da educação e da formação em Big Data para atender à crescente procura por talento analítico profundo e capacidade analítica para a força de trabalho em geral*” (Kurose & Marzullo, 2016). Para a IBM (2017), existem dez novas carreiras na área de BD, para além da tradicional carreira de administrador de bases de dados, e, de entre as capacidades exigidas, estão a modelação de BD e o conhecimento da linguagem SQL (Markow et al., 2017).

As bases de dados relacionais, conforme ranking do DB-Engines (2019a), continuam no topo dos Sistemas de Gestão de Bases de Dados (SGBD) em uso. Nem mesmo todo o volume de dados gerado anualmente, somado com as novas tecnologias de armazenamento de dados não relacionais, foram capazes de fazer frente à tradicional abordagem relacional, que tem quase meio século. O sucesso do modelo relacional deve-se aos esforços de investigações anteriores e, pelo que tudo indica, dificilmente será substituído a curto prazo.

As bases de dados relacionais têm grande relevância para a comunidade acadêmica, em especial para a área da informática, uma vez que os cursos de licenciatura possuem um componente obrigatório com foco na aprendizagem neste domínio. Essa disciplina faz parte das Diretrizes do Programa de Licenciatura recomendadas pela *Association for Computing Machinery* (ACM) e *IEEE Computer Society* para os cursos de Informática e Tecnologia (ACM & IEEE, 2013). Entre os tópicos da disciplina estão os fundamentos de projeto e modelação de bases de dados, que resultam num diagrama que representa a estrutura da BD. Os diagramas são cada vez mais usados em muitos métodos de projeto e são ensinados em vários contextos no ensino superior, como projeto de BD conceptual ou projeto de engenharia de software. Eles são uma parte importante de muitas avaliações (Hussain, 2016).

No contexto educacional, a modelação de BD é uma tarefa complexa. De acordo com Hussain (2016), entre os motivos que levam os alunos a cometerem erros durante o processo de modelação, destacam-se as premissas erróneas e a falta de pressupostos necessários. Uma outra dificuldade com que os alunos se deparam é que, dado um conjunto de entidades, existem potencialmente grandes números de relações possíveis (D. Batra & Antony, 1994; Dinesh Batra, 2007). Até mesmo com auxílio computadorizado, os alunos precisam de no mínimo 20 minutos para solucionar um problema de modelo ER (Elmadani, Mitrovic, Weerasinghe, & Neshatian, 2015). Isso ocorre porque a subjetividade é inerente ao processo de modelação, uma vez que, para projetar uma solução, os alunos são submetidos a uma tarefa de interpretação de um problema de um domínio qualquer que é usualmente descrito em linguagem natural. Por outro lado, é praticamente impossível ao professor fornecer uma avaliação personalizada dentro de sala, devido às mais variadas interpretações e respostas que os alunos fornecem. Entretanto, isso não seria impossível para o computador, se um sistema fosse capaz de automatizar o processo de avaliação de diagramas de entidade e relacionamento (DER), o professor poderia saber quais são os conceitos de BD que os alunos dominam ou que estejam a precisar de apoio. Para o aluno, seria extremamente benéfico ter um sistema de avaliação automática com mecanismos de aprendizagem autónoma, através dos mais variados *feedbacks*.

Na educação, os diagramas são explorados como ferramentas que melhoram a aprendizagem (Cheng, Lowe, & Scaife, 2001; L. Yang & Cao, 2016). Especialmente quando se trata de diagramas de entidade e relacionamento, a avaliação do DER oferece desafios que atraem o interesse da investigação na avaliação automática, tais como:

- 1) Ambiguidade: existem pelo menos dois tipos de ambiguidade: conceptual e na especificação de requisito. A conceptual é aquela em que os autores divergem sobre um mesmo conceito de DER (Pendegraft, 2012). De outro modo, a ambiguidade na especificação de requisito ocorre na descrição do enunciado de uma questão de modelação, que é usualmente fornecido em linguagem natural para o aluno interpretar e fornecer uma solução em DER, e podem ter um foco muito abrangente de soluções (I. Y. Song, Zhu, Ceong, & Thonggoom, 2015);
- 2) Complexidade combinatória: estudos recentes e anteriores confirmam que alunos iniciantes têm dificuldade em modelar todos os tipos de relacionamentos, incluindo relacionamentos binários. Esse problema deve-se às diferentes maneiras de conectar duas entidades a um relacionamento, por exemplo, se existem cinco entidades, a solução pode ter C_2^5 ou dez relacionamentos binários, ou pode ter C_3^5 ou dez relacionamentos ternários, e assim por diante (D. Batra & Antony, 1994; Dinesh Batra, 2007; Hussain, 2016);
- 3) Falha de correspondência semântica: a tradução literal da especificação de requisitos pode levar ao erro, pois nem todas as relações do mundo real mapeiam relações de bases de dados, e algumas relações são deriváveis no nível de bases de dados (Dinesh Batra, 2007);
- 4) Inexperiência e conhecimento incompleto de projetistas: é difícil até mesmo para um especialista identificar as entidades ocultas, pois são necessárias experiências de domínio para conseguir criar um modelo completo, com experiências passadas (Kim, Lee, & Moon, 2008);
- 5) Regras de modelação dispersas: existem várias regras heurísticas do modelo ER, por exemplo “Um substantivo comum corresponde a um tipo de entidade num diagrama de ER.” (Hartmann & Link, 2007; Lucassen, Robeer, Dalpiaz, van der Werf, & Brinkkemper, 2017), mas uma entidade também pode ser identificada a partir de um verbo e requisitos implícitos (I.-Y. Song, Yano, Trujillo, & Lujan-Mora, 2011). Essas regras são uteis, porém são consideradas incompletas e conflituais e podem fornecer uma recomendação errada e levar a erros cognitivos (I. Y. Song et al., 2015).

- 6) Múltiplas soluções corretas versus solução única: aceitar apenas uma única solução de DER correta implica classificar uma solução DER em certa ou errada, enquanto de facto existe um grande número de DER alternativos que podem ser criados para um problema específico (D. L. Moody & Shanks, 1994), e isto envolve um problema de equivalência entre DER (D. L. Moody, 1998; D. L. Moody & Shanks, 2003).

A avaliação automática de soluções diagramáticas não é um campo relativamente novo, tendo mais de vinte anos (Hall & Gordon, 1998). No entanto, uma revisão da literatura sobre as soluções para avaliação automática de DER revela que esta área não pode ser considerada concluída, pois apresenta uma série de limitações que podem ser superadas, dentre as quais se destacam, **a) Interface restritiva:** sistemas que não permitem ao utilizador desenhar o DER usando as funções do rato de clicar e arrastar; **b) Resposta restritiva:** são sistemas que aceitam apenas uma solução correta e forçam os alunos a desenvolver uma resposta padrão através da interface gráfica (usualmente o sistema fornece um *feedback* do tipo certo ou errado); **c) Destaques visuais:** são sistemas que usam realces visuais e simplificam o enunciado da questão, fornecendo a resposta esperada do sistema com um conjunto de palavras destacadas em diferentes cores; **d) Notação restritiva:** são sistemas que oferecem uma notação desconhecida ou sem todas as propriedades do DER, reduzindo assim os elementos de avaliação do DER e o *feedback* do sistema; **e) Avaliação sem Equivalência:** são sistemas que aceitam apenas uma resposta única ou que não tratam a equivalência entre os DER; **f) Feedback inadequado:** são sistemas que fornecem aos alunos um *feedback* positivo para respostas de DER incorreto; **g) Feedback estático:** são sistemas que usam uma abordagem limitada com um conjunto de regras pré-estabelecidas para fornecer o *feedback*, possuindo portanto um *feedback* estático para casos de deteção e correção de erros.

As ferramentas de engenharia de software assistida por computador (CASE) também são usadas para facilitar o processo de ensino de BD. Normalmente essas ferramentas são empregues para converter um modelo de dados conceptual num modelo físico de implementação (L. Yang & Cao, 2016). Nessa linha, muitos investigadores propuseram sistemas para projetar o DER, mas sem avaliação automática (Antony & Batra, 2002; Bogdanović, Stanimirović, Davidović, & Stoimenov, 2008; Dimitrieski, Ćeliković, Aleksic, Ristić, & Luković, 2014; González & Suthers, 2000). A principal limitação dessas ferramentas propostas é que elas não são capazes de entender as respostas dos alunos e enviar

um *feedback* acerca do problema. Portanto, as ferramentas CASE não podem ser utilizadas no processo de ensino para corrigir as respostas dos alunos e auxiliar o professor nas avaliações.

As ferramentas encontradas na literatura para avaliar automaticamente DER são baseadas em tutores inteligentes (Mitrovic & Suraweera, 2016), processamento de linguagem natural (Thomas, 2013) e regras heurísticas (Bligh, 2007; Hall & Gordon, 1998). Entretanto, apesar das ferramentas propostas conseguirem avaliar o DER, elas usam alguns recursos tecnológicos para não demonstrar as fraquezas das abordagens. Algumas soluções usam notações reduzidas, ou até mesmo desconhecidas, para evitar a complexidade de avaliar a variedade de itens existentes no DER, e assim, não avaliam adequadamente as propriedades do diagrama. Outras ferramentas oferecem um conjunto de palavras que fazem parte da solução esperada, e assim evitam a análise de equivalência entre DER. Essas abordagens são ineficientes na análise do DER para detecção e correção de erros, e as respostas dos alunos são classificadas em certo ou errado. Não permitindo mais de uma resposta por questão, estas abordagens obrigam o aluno a parar de responder e não incentivam a submeter novas respostas.

1.2 Objetivos

A investigação realizada teve como objetivos: (i) identificar as abordagens de avaliação de diagramas de entidade e relacionamento; (ii) propor um ambiente virtual de ensino e aprendizagem para avaliar automaticamente diagramas de entidade e relacionamento no contexto de cursos de bases de dados relacionais. Conseqüentemente a questão principal desta investigação foi:

“Como se pode automatizar a avaliação de DER no contexto de ensino e aprendizagem de Bases de Dados Relacionais?”

Para podermos explorar as características de avaliação automática de DER e operacionalizar a investigação e responder à questão, decidimos subdividi-la em três novas questões. Na primeira questão desejamos reconhecer o contexto de aplicação de avaliação de DER:

Q1 - Quais são as abordagens para avaliar os DER e como são utilizadas?

A resposta a esta questão pode-nos revelar: (i) quais são os elementos que devem compor uma avaliação do DER; (ii) como avaliar as diferentes propriedades de um DER; (iii) esclarecer até que ponto as técnicas existentes, processadas por computador, conseguem medir os conceitos e elementos que formam o DER.

Uma vez que identificamos as características do DER e a sua mecânica de avaliação, podemos investigar aspetos específicos dos ambientes virtuais que se propõem avaliar os diagramas, e assim chegamos à segunda questão:

Q2 - Como os DER são avaliados nos ambientes virtuais de aprendizagem?

Ao responder a esta questão, pretendemos identificar: (i) quais são os ambientes e os métodos usados na avaliação DER; (ii) quais os mecanismos tecnológicos, seja através de elementos visuais ou não, em que o aluno fornece uma solução; (iii) em que condições, seja em avaliações formativas ou sumativas, os alunos e professores podem usar a avaliação automática; (iv) quais são os tipos de *feedback* e como os alunos e os professores os utilizam; (v) quais são as lacunas desses ambientes.

Após uma ampla revisão de literatura sobre os ambientes virtuais de BD e de identificarmos as forças e fraquezas relacionadas à avaliação de DER, podemos investigar diferentes estratégias para fechar as lacunas e inovar no processo de avaliação automática de DER, e assim chegamos à terceira questão:

Q3 - Como integrar diferentes técnicas, processadas por computador, para avaliar o DER e fornecer um *feedback* mais próximo do professor?

Ao responder a esta questão queremos definir (i) um método de avaliação que integra técnicas de compiladores, métricas de engenharia de software e aprendizagem computacional supervisionada para fornecer *feedback*; (ii) um modelo de avaliação de DER mais próxima da avaliação do professor; (iii) indicar os algoritmos de aprendizagem computacional que são mais eficazes e ineficazes na construção de modelos preditivos de DER; (iv) um conjunto de características de DER que melhor representem a avaliação do professor. (v) uma abordagem de avaliação que possa ser usada nas modalidades de avaliação formativa e sumativa; (vi) um sistema que supere as lacunas detetadas nas demais ferramentas de avaliação automática de DER.

Em conjunto, estas três questões direcionam a resposta à questão principal, identificando como se avalia um DER (questão 1), como os DER são avaliados em ambientes

virtuais de avaliação de DER (questão 2) e como usá-los de maneira integrada num sistema computacional no contexto da disciplina de bases de dados relacionais (questão 3).

1.3 Metodologia de Investigação

Esta secção apresenta a *Design Science Research Methodology* (DSRM) como método de investigação aplicado neste trabalho e como ele guiou a realização desta investigação (Peppers, Tuunanen, Rothenberger, & Chatterjee, 2007).

A investigação no campo de sistemas de informação caracteriza-se por trabalhar com diferentes resultados, que são comumente chamados de artefactos. Os artefactos de Tecnologia da Informação (TI) são considerados construções artificiais, i.e., desenvolvidas por humanos. Tais artefactos são habitualmente o objetivo final de qualquer projeto de investigação em sistemas de informação e podem ser exemplificados como um sistema informático, modelo, *framework*, algoritmo, protótipos, processos ou outras soluções tecnológicas (Hevner & Chatterjee, 2010).

A complexidade de escolher uma metodologia de investigação na área de sistemas de informação deve-se ao facto de a metodologia ser capaz de abrigar não só diferentes artefactos, mas diferentes métodos e técnicas, dando liberdade ao investigador para aplicar diferentes abordagens conforme as interações de avaliação e melhoria da solução.

A partir de uma revisão de várias peças de literatura sobre processos de investigação nas áreas de engenharia, informática, e sistemas de informação, Peppers et al. (2007) observou que o foco da investigação é dado ao projeto do artefacto, isto é, ao processo de produção do artefacto. Peppers et al. (2007) usou as principais contribuições de várias publicações para criar um modelo de processo generalizado para investigação em sistemas de informação. Elas concentraram-se nas similaridades das visões de vários investigadores no campo para criar o modelo de processo conhecido como DSRM.

O modelo de processo DSRM de Peppers (2007) caracteriza-se pela sequência de seis etapas, começando pela identificação do problema e a motivação, seguidas de um processo iterativo de definição dos objetivos, projeto e desenvolvimento, demonstração, avaliação e comunicação. Cada etapa da metodologia DSRM é descrita a seguir e a Figura 1-1 ilustra como cada uma foi abrangida por esta investigação:

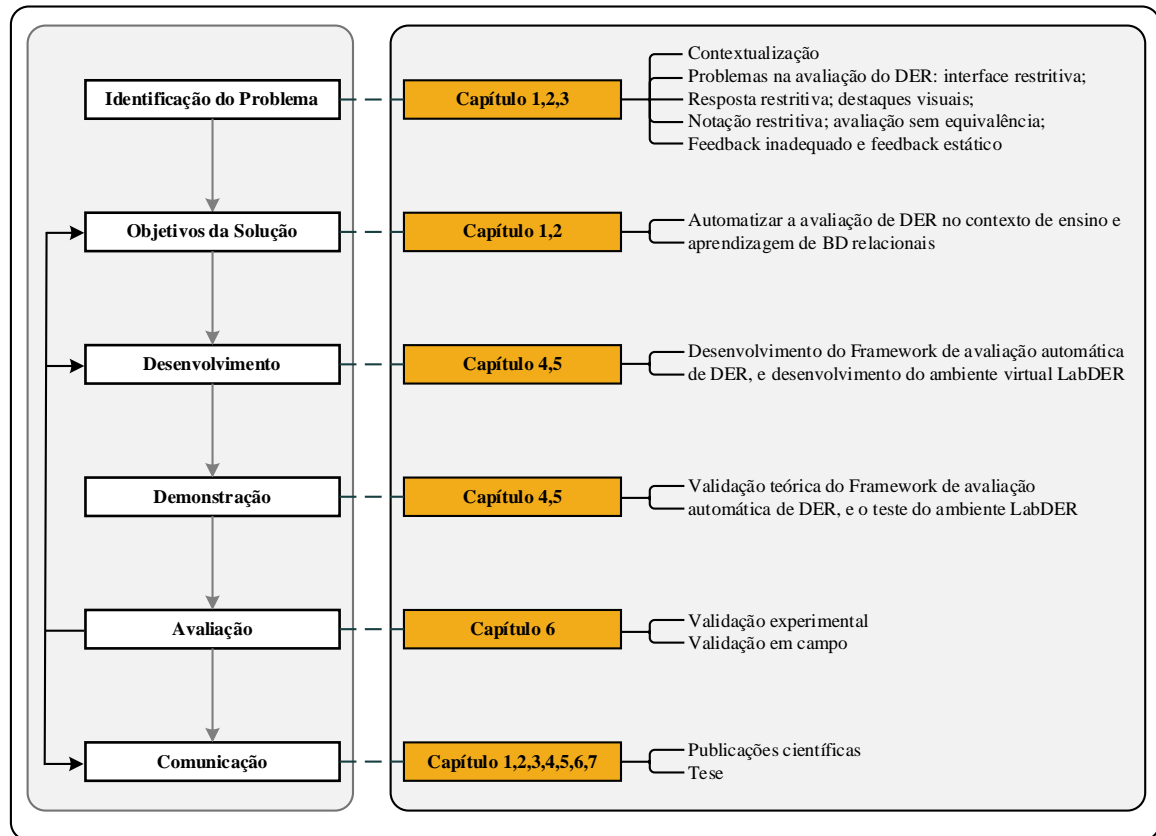


Figura 1-1. Método DSRM de investigação da tese.

- 1. Identificação do problema:** A falta de um sistema informático que avalie automaticamente as questões de DER é um problema significativo no contexto ensino e aprendizagem de bases de dados relacionais. Os atuais ambientes de aprendizagem sobre DER não tiveram a aceitação esperada pelos utilizadores, muito em detrimento do ambiente ser restritivo para desenho do diagrama e por dispor de um processo de avaliação manual que sobrecarrega o professor. Vários modelos de avaliação de ER têm sido estudados, com a intenção de melhorar o processo de aprendizagem através de ambiente online. No entanto, muitos desses projetos não ofereceram as melhorias esperadas e muitos foram descontinuados. O capítulo 2 discute os métodos de avaliação de DER e as limitações dessas abordagens. Portanto, a primeira etapa realiza uma análise pormenorizada sobre as restrições de funcionalidade, restrições tecnológicas e de domínio para classificá-las em várias categorias.
- 2. Objetivos da Solução:** Nesta etapa busca-se definir os objetivos que devem ser inferidos racionalmente a partir da especificação do problema. Nesta investigação, o foco está no modelo de avaliação automática de DER para um ambiente virtual de ensino e aprendizagem de BD. Devido a complexidade do contexto de desenvolvimento do

produto final, é fundamental alcançar uma série de objetivos que juntos irão compor o todo. Para cada objetivo, foram aplicadas técnicas específicas de acordo com o tipo de artefacto a fim de alcançar além do resultado a qualidade esperada. Os objetivos estão na secção 1.2.

3. **Desenvolvimento:** Esta etapa é dedicada à criação do artefacto em si, inclui o desenvolvimento do *framework* de avaliação automática de DER e SQL e o ambiente virtual de aprendizagem LabDER, discutido no capítulo 4 e 5. Essa fase inclui todos os processos, técnicas, métodos, e procedimentos relacionados ao desenvolvimento dos objetivos.
4. **Demonstração:** Esta etapa demonstra a aplicabilidade dos artefactos projetados para provar que os produtos desenvolvidos resolvem o problema. Isto inclui, a validação teórica do *Framework* de avaliação automática de DER e SQL (capítulo 4), e o teste do ambiente LabDER através de simulações (capítulo 5).
5. **Avaliação:** Essa etapa envolve observar e medir o quão bem o artefacto atinge os objetivos. Foi possível comprovar que o modelo de avaliação automática de DER estava de acordo com os objetivos pressupostos observando os resultados reais do uso do artefacto por meio de métodos, métricas de aprendizagem computacional, análises quantitativas, validação experimental e validação em campo.
6. **Comunicação:** O resultado da investigação é outra etapa da DSRM. Todas as análises e medidas propostas foram publicadas em conferências relevantes e periódicos de alto impacto na área (ver Publicações resultantes deste trabalho de investigação).

Esta tese também pode ser vista como um processo de três etapas que exercem um papel fundamental na concepção do todo.

1ª etapa) Desenvolvimento do LabDER: onde o principal resultado foi um ambiente virtual de aprendizagem que pudesse ser utilizado em sala de aula para recolha de dados (diagrama ER). Para desenvolver o LabDER, foram adotados um processo de desenvolvimento de software e um processo de teste para a sua validação.

2ª etapa) Desenvolvimento do *Framework*: além da revisão da literatura, foram necessários vários testes para ajudar a desenhar a melhor abordagem para o cálculo de todas as variáveis necessárias para a recolha dos dados.

3ª etapa) Modelo de ML: O principal resultado foi identificar o modelo de avaliação de ER que mais se aproxima da avaliação do professor. Mas, para isso, foi utilizado um processo de transformação de dados para coleta e processamento dos dados.

Estas três fases estão descritas nos capítulos 4, 5 e 6.

1.4 Contributos

As principais contribuições desta investigação são:

1. Uma revisão da literatura que identificou as abordagens atuais e mais comuns usadas na avaliação de DER, a qual analisou as ferramentas educacionais de avaliação automática de DER com destaque para as suas vantagens e desvantagens em cada abordagem;
2. Um *framework* inovador inspirado em compiladores, métricas de engenharia de software e aprendizagem computacional supervisionada para avaliar automaticamente o diagrama de entidade e relacionamento;
 - a. O *framework* fornece um *feedback* semântico, que auxilia o aluno na deteção e correção do erro, e oferece ao aluno a opção de refazer a solução. O *feedback* semântico inclui:
 - i. Uma análise léxica, sintática e semântica que identifica cada uma das propriedades do DER da resposta modelo e do aluno.
 - ii. Uma lista de métricas que calculam as distâncias entre as propriedades do DER do aluno e do professor, e fornece um valor acerca dos conceitos de BD ausentes ou presentes.
 - iii. Um modelo preditivo, baseado em algoritmos de aprendizagem computacional supervisionada, fornece a nota final do aluno.
 - b. O *framework* proposto é generalizável, pode ser usado para uma variedade de tecnologias de soluções diagramáticas e código fonte.
3. Um ambiente virtual de ensino de bases de dados relacionais, denominado LabDER, com os seguintes aspetos:
 - a. A partir de um processo de desenvolvimento de software, um sistema educacional foi projetado, desenvolvido e testado para garantir o adequado

funcionamento em cursos de bases de dados. O sistema pode ser usado para avaliar questões de DER, SQL, de escolha múltipla e de texto livre nas modalidades de avaliação formativa e sumativa.

- b. Processos de testes de software simplificados foram projetados para avaliar a funcionalidade, desempenho e usabilidade do LabDER. Esses processos de testes seguiram as recomendações mais apropriadas tendo em conta a abrangência e o contexto do projeto.
- c. O LabDER foi implementado em seis universidades, localizadas em vários locais do mundo. O sistema foi usado por 330 utilizadores, divididos em seis turmas, que submeteram mais de 40 mil respostas.

1.5 Estrutura da tese

A tese está organizada em seis capítulos. O capítulo 1 apresenta uma breve descrição dos objetivos, do problema abordado nas questões de investigação, a metodologia de investigação seguida, os contributos e a organização da tese. O capítulo 2 apresenta a base teórica para o desenvolvimento do avaliador automático, descrevendo os aspetos de interesse de avaliação do DER e os desafios. O capítulo 3 apresenta uma análise das ferramentas que fazem a avaliação de DER, encontradas na literatura, destacando as abordagens utilizadas e as limitações. O capítulo 4 descreve brevemente como a estrutura do *framework* de avaliação automática nasceu originalmente para avaliar código fonte SQL e evoluiu para avaliação automática de diagramas ER. Também apresenta como são compostas as estruturas do *framework* e como se avalia automaticamente DER. O capítulo 5 apresenta o processo de desenvolvimento do ambiente virtual para o ensino de bases de dados relacionais e descreve os resultados dos testes de funcionalidade, desempenho e usabilidade para validar a ferramenta antes de entrar em produção. O capítulo 6 descreve o processo de implantação do ambiente nas universidades e mostra os resultados da investigação, destacando os elementos que reconhecemos terem importância nos ensaios. Também descreve a avaliação do ponto de vista dos educadores e documenta os resultados com os alunos exemplos. O último capítulo analisa as principais questões de investigação e aponta suas principais contribuições e finaliza com sugestões para trabalhos futuros.

Fundamentação teórica

Este capítulo visa relatar a teoria que serviu de base à presente investigação. Foi estruturado em modelo ER e diagrama ER. A teoria apresentada neste capítulo é usada no decorrer da tese, entretanto não substitui os livros de BD que descrevem pormenorizadamente tais teorias.

Portanto, este capítulo descreve os conceitos de diagramas ER no contexto de ensino de BD, bem como o ponto de vista das limitações dessa abordagem descrita na literatura.

2.1 Modelo ER

A tecnologia mais amplamente adotada pelas empresas para armazenar dados é baseada no modelo relacional, que surgiu em meados da década de 70 (Teorey, Lightstone, Nadeau, & Jagadish, 2011). O site DB-Engines (2019) apresenta o modelo relacional como a tecnologia mais largamente empregada no armazenamento de dados (DB-Engines, 2019a) e (DB-Engines, 2019b), sendo esta tecnologia implementada pelos Sistemas de Gestão de Bases de Dados (SGBD). Mesmo com a evolução da tecnologia de bases de dados com sistemas especializados (multimídia, espaciais, orientados a objetos, entre outros), os SGBD relacionais continuam a ser a tecnologia dominante nas empresas.

Segundo Teorey et al. (2011), o modelo ER é o processo de representar visualmente entidades, atributos e relacionamentos para produzir um diagrama de entidade e relacionamento (DER). O processo é de natureza repetitiva uma vez que as entidades são descobertas ao longo do projeto de bases de dados. Feito corretamente, o DER é independente da plataforma e até pode ser usado por bases de dados não relacionais (Teorey et al., 2011).

Conforme Speelpenning, Daux e Gallus (2001), um modelo de entidade e relacionamento (modelo ER) é “uma lista de todas as entidades e atributos, bem como todos os relacionamentos entre as entidades que são importantes.” (Speelpenning, Daux, & Gallus, 2001). A modelação de dados é útil para mapear os significados e interações sobre domínios do mundo real num

esquema conceptual, e requer três conceitos básicos: conjuntos de entidades, conjuntos de relacionamentos, e atributos (Elmasri & Navathe, 2015). Esses conceitos são considerados os elementos essenciais para entendimento e avaliação do DER.

Uma entidade é uma pessoa, lugar, coisa ou evento sobre o qual os dados serão recolhidos e armazenados. Uma entidade representa um tipo particular de objeto no mundo real, o que significa que uma entidade é “distinguível” - isto é, a ocorrência de cada entidade é única e distinta. Por exemplo, uma entidade CLIENTE teria muitas ocorrências distintas de clientes, como João Silva, Pedro Souza e Joaquim Pereira. As entidades podem ser objetos físicos, como clientes ou produtos, mas entidades também podem ser abstrações, como rotas de voo ou espetáculos musicais (Coronel & Morris, 2019).

Um atributo é uma característica de uma entidade. Por exemplo, uma entidade CLIENTE seria descrita por atributos como sobrenome do cliente, nome do cliente, número de telefone do cliente, endereço do cliente e limite de crédito do cliente. Atributos são o equivalente de campos em sistemas de arquivos (Coronel & Morris, 2019).

Um relacionamento descreve uma associação entre entidades. Por exemplo, existe uma relação entre clientes e agentes que pode ser descrita da seguinte maneira: um agente pode atender muitos clientes e cada cliente pode ser atendido por um agente. Os modelos de dados usam três tipos de relacionamentos: um-para-muitos, muitos-para-muitos e um-para-um. Nas ferramentas de modelo ER, geralmente usam-se notações abreviadas 1: M ou 1 .. *, M: N ou * .. * e 1: 1 ou 1..1, respectivamente. Embora a notação M: N seja um rótulo padrão para o relacionamento muitos-para-muitos, o rótulo M: M também pode ser usado. Em algumas ferramentas de modelo ER é utilizado o conceito de restrição, que é uma restrição colocada nos dados. As restrições são importantes porque ajudam a garantir a integridade dos dados. As restrições são normalmente expressas sob a forma de regras (Coronel & Morris, 2019).

No presente, a modelação é parcialmente implementável por um conjunto de software que permite desenhar os DER. Esses softwares podem aparecer como um componente integrado ao SGBD, ou em ferramentas de engenharia de software auxiliada por computador (CASE). Ferramentas CASE usualmente aceitam a concepção do DER na notação Engenharia da Informação e IDEF1X, tanto ao nível conceptual, como o lógico e o físico, como é o caso das ferramentas ERwin Data Modeler e Visual Paradigm. De forma análoga, os componentes de desenho que estão integrados ao SGBD, oferecem uma notação exclusiva, como por exemplo os da Microsoft e Oracle. As notações integradas aos SGBD são inspiradas na notação engenharia da informação, que apresenta visualmente as tabelas em formas de retângulos, e as

relações através de linhas. Na versão mais recente do SGBD MySQL é fornecido uma ferramenta denominada MySQL Workbench, que faz uso da notação engenharia da informação para representar o DER.

A MER forma a base do DER que representa a base de dados conceptual conforme visualizado pelo utilizador final (Coronel & Morris, 2019). Entretanto, existem características do modelo que são representadas de forma diferente, dependendo da notação utilizada.

2.2 Diagramas de Entidade e Relacionamento

Um diagrama de entidade e relacionamento ou diagrama ER é uma figura que usa representações gráficas para modelar os componentes da base de dados (Coronel & Morris, 2019). Normalmente, um modelo é representado em vários diagramas, mostrando diferentes perspetivas de negócios (Speelpenning et al., 2001).

Os diagramas ER são usados para modelar e projetar bases de dados relacionais, em termos de lógica, regras de negócio e tecnologia (Coronel & Morris, 2019). A modelação dos dados pode ser representada em três diferentes níveis de abstração:

- modelo de dados conceptual - providencia uma visão do utilizador do que deve ser armazenado na BD. Inclui as entidades, relacionamentos, processos e restrições apropriados, impostos pelo negócio;
- modelo de dados lógico - providencia uma visão global dos principais objetos de dados, independente de software e hardware. Ou seja, não especifica detalhes do modelo de BD, como o tipo de dado do atributo;
- modelo de dados físico - providencia uma representação do esquema interno que deve mapear o modelo conceptual para as construções do modelo relacional. Portanto, as entidades do modelo conceptual são mapeadas para tabelas no modelo relacional, assim como atributos em colunas, e adicionadas todas as informações para implementação do nível de SGBD, como tipo de dados, índices, entre outros. O modelo físico pode ser expresso em SQL, a linguagem padrão para bases de dados relacionais.

A Tabela 2-1 apresenta os três níveis de abstração dos dados, conceptual, lógico e físico, e uma representação visual correspondente ao seu modelo, baseado no livro de Coronel e Morris (2019). O modelo conceptual identifica em alto nível o relacionamento entre as

entidades, portanto incluem somente as entidades mais importantes e os seus relacionamentos, sem especificação de chaves e atributos. O modelo lógico identifica em maior detalhe, sem incluir informações da implementação física na base de dados, portanto, este modelo inclui as entidades, os atributos, as chaves primárias e estrangeiras. O modelo físico representa como os dados serão implementados internamente na BD relacional. Ou seja, o DER deve especificar todas as tabelas, colunas, chaves primárias, chaves estrangeiras, tipos de dados, se a coluna é requerida e todas as especificações necessárias para definir a BD.

Tabela 2-1. Níveis de abstração do dado, baseado em (Coronel & Morris, 2019).

Modelo	Representação visual						
Conceptual							
Lógico	<table border="1"> <thead> <tr> <th>Departamento</th> <th>Curso</th> </tr> </thead> <tbody> <tr> <td>PK <u>Dept_Cod</u></td> <td>PK <u>Cur_Cod</u></td> </tr> <tr> <td>Dept_Nome</td> <td>FK1 Dept_Cod Cur_Titulo Cur_Descricao Cur_Credito</td> </tr> </tbody> </table>	Departamento	Curso	PK <u>Dept_Cod</u>	PK <u>Cur_Cod</u>	Dept_Nome	FK1 Dept_Cod Cur_Titulo Cur_Descricao Cur_Credito
Departamento	Curso						
PK <u>Dept_Cod</u>	PK <u>Cur_Cod</u>						
Dept_Nome	FK1 Dept_Cod Cur_Titulo Cur_Descricao Cur_Credito						
Físico	<table border="1"> <thead> <tr> <th>Departamento</th> <th>Curso</th> </tr> </thead> <tbody> <tr> <td>PK <u>Dept_Cod</u> INT</td> <td>PK <u>Cur_Cod</u> INT</td> </tr> <tr> <td>Dept_Nome CHAR(30)</td> <td>FK1 Dept_Cod INT Cur_Titulo CHAR(30) Cur_Descricao CHAR(100) Cur_Credito INT</td> </tr> </tbody> </table>	Departamento	Curso	PK <u>Dept_Cod</u> INT	PK <u>Cur_Cod</u> INT	Dept_Nome CHAR(30)	FK1 Dept_Cod INT Cur_Titulo CHAR(30) Cur_Descricao CHAR(100) Cur_Credito INT
Departamento	Curso						
PK <u>Dept_Cod</u> INT	PK <u>Cur_Cod</u> INT						
Dept_Nome CHAR(30)	FK1 Dept_Cod INT Cur_Titulo CHAR(30) Cur_Descricao CHAR(100) Cur_Credito INT						

Do ponto de vista de ensino de BD, o modelo físico é extremamente útil, porque reúne diversos conceitos visualmente e tem correspondência direta com a linguagem SQL, e, portanto, pode ser utilizado para criar estruturas de BD no SGBD. Estas características tornam o modelo físico mais atrativo para aprofundar o conhecimento de BD, visto que os conceitos têm aplicação prática. Esses mesmos conceitos de BD são a base teórica para compor um DER básico, e estão listados na Tabela 2-2. Outros conceitos importantes que também fazem parte da base teórica do ensino de BD como restrição de integridade, índices, e normalização poderiam ser incluídos nessa listagem. Entretanto, esses conceitos são suficientes para balizar os itens desejáveis de avaliação de DER no ensino de BD.

Tabela 2-2. Principais conceitos de BD usados no DER, extraído de (Coronel & Morris, 2019).

Elementos	Descrição
Tabela	Uma Tabela representa um tipo particular de objeto no mundo real, o que significa que uma tabela é “distinguível” - isto é, a ocorrência de cada tabela é única e distinta. Cada tabela consiste num conjunto de atributos que descreve características particulares da tabela.
Colunas	Uma coluna representa características comuns da tabela. Cada coluna tem um nome distinto.
Colunas requeridas	É uma coluna que deve ter um valor.
Colunas opcionais	É uma coluna que não requer um valor; portanto, pode ser deixado vazio.
Domínio da coluna	Cada coluna possui um intervalo específico de valores conhecido como domínio de atributo. Todos domínios conhecidos no MySQL são mapeados em quatro classes: numérico, caráter, data e miscelâneas.
Chave primária	Um atributo que identifica exclusivamente cada instância da tabela, garante que os valores são únicos e não pode conter valores nulos.
Chave Primária Composta	Um conjunto de atributos que identificam exclusivamente cada instância da entidade.
Relacionamento	Descreve uma associação entre entidades, são três tipos de relacionamentos: um-para-muitos (1:M), muitos-para-muitos (M:N) e um-para-um (1:1).
Relacionamento fraco	É um relacionamento não identificador, existe se a chave primária da entidade relacionada não contiver um componente de chave primária da entidade pai.
Relacionamento forte	É um relacionamento de identificação, existe quando a chave primária da entidade relacionada contém um componente de chave primária da entidade pai.
Grau do relacionamento	Indica o número de entidades ou participantes associados no relacionamento. Um relacionamento unário é mantido quando existe uma associação dentro de uma única entidade. O relacionamento binário existe quando duas entidades estão associadas. Um relacionamento ternário existe quando três entidades estão associadas. A denominação N-Ary representa as associações entre quatro mais entidades.
Cardinalidade	Número mínimo e máximo de ocorrências de entidade associadas a uma ocorrência da entidade relacionada.
Participação	A participação num relacionamento de entidade é opcional ou obrigatória. Participação opcional significa que uma ocorrência de entidade não requer uma ocorrência de entidade correspondente num relacionamento específico. Participação obrigatória significa que uma ocorrência de entidade requer uma ocorrência de entidade correspondente num relacionamento particular.
Especialização/Generalização	Enfatiza a semelhança entre diferentes tipos de entidades e abstrai as suas diferenças. Subdivide entidades semelhantes em conjuntos de entidades mais específicas e enfatiza as suas diferenças. As relações representadas dentro da hierarquia de especialização são algumas vezes descritas em termos de relacionamentos “é-um”. Ao nível de implementação, o supertipo e o (s) seu (s) subtipo (s) descritos na hierarquia de especialização mantêm uma relação de 1: 1.

O padrão ISO/IEC/IEEE 31320-2:2012 define as normas de modelação de dados seguindo a notação IDEF1X (ISO/IEC/IEEE, 2012). Este padrão descreve também os três níveis de modelação, e faz uma comparação entre os modelos pelo mapeamento dos elementos conceituados de cada nível. A Tabela 2-3 resume esses níveis de modelação propostos originalmente no padrão ISO/IEC/IEEE 31320-2:2012, com adaptações de terminologias utilizadas no livro de Carlos Coronel e Morris (2019) descrita na Tabela 2-2.

Tabela 2-3. Níveis de modelo por conceitos de BD, uma adaptação dos níveis de estilo de construções do IDEF1X (ISO/IEC/IEEE, 2012).

Elementos	Modelo Conceptual	Modelo Lógico	Modelo Físico
Tabela	Sim	Sim, tipicamente abstrato	Sim
Colunas	Não	Sim, tipicamente abstrato	Sim
Colunas requeridas	Não	Sim	Sim
Colunas opcionais	Não	Sim	Sim
Domínio da coluna	Não	Não	Sim
Chave primária	Não	Sim	Sim
Chave Primária Composta	Não	Sim	Sim
Relacionamento	Sim	Sim	Sim
Relacionamento fraco	Não	Algun	Sim
Relacionamento forte	Não	Algun	Sim
Grau do relacionamento	Não	Algun	Sim
Cardinalidade	Não	Algun	Sim
Participação	Não	Algun	Sim
Especialização/ Generalização	Não	Sim	Sim

O modelo físico é o mais completo e pode ser implementado diretamente na BD. A principal diferença entre o modelo conceptual e o físico reside no grau de abstração dada a um determinado conceito. Por exemplo, no modelo físico, uma coluna deveria seguir um padrão de nomenclatura, para evitar o problema de homónimos, e ao mesmo tempo garantir a singularidade da coluna numa tabela. Porém, a coluna no modelo conceptual reflete em termos de semântica (o significado da propriedade) e sintaxe (a assinatura) na regra de negócio. Ou seja, o modelo conceptual preocupa-se em modelar para promover a resiliência do sistema diante das mudanças, em vez de propriedades individuais (ISO/IEC/IEEE, 2012).

Conforme descrito no padrão ISO/IEC/IEEE 31320-2:2012, em alguns casos, a transformação do modelo conceptual para o físico pode não produzir a melhor implementação, e podem ser feitas alterações por motivos de desempenho, disponibilidade, capacidade de manutenção ou operacionais (ISO/IEC/IEEE, 2012). Essas alterações são sempre representadas no DER físico, seja por exemplo, um suporte na interface do utilizador que tenha tabelas adicionais na BD, ou em ambientes distribuídos, onde as tabelas também podem ser particionadas numa série de BD.

Portanto, o modelo físico possui todas as informações necessárias para implementar uma visualização específica para uma determinada plataforma. Por exemplo, no caso da plataforma relacional, a visualização será em termos de tabelas, colunas, tipos de dados, restrições de integridade referencial e assim por diante (ISO/IEC/IEEE, 2012).

Atualmente, existem vários estilos de notações dos diagramas ER, entre elas, as mais comumente usadas são de Chen, *Crow's foot*, e IDEF1X (Coronel & Morris, 2019). A seguir, estas notações são descritas para que se possa contrastar detalhes específicos. As notações escolhidas foram selecionadas com base no seu amplo uso em textos acadêmicos, periódicos e ferramentas CASE.

2.2.1 Notações dos diagramas de entidade e relacionamento

Existem algumas variações de formatos de representação dos diagramas de entidade e relacionamento. Peter Chen desenvolveu o formato DER em 1976 (P. P.-S. Chen & Pin-Shan, 1976), desde então, vendedores de software, cientistas de informática e acadêmicos desenvolveram muitas variações, todas conceptualmente iguais, como IDEF1X (NIST, 1993) e *Crow's foot* (Bachman, 1969). Entre esses estilos de notação, as tabelas são representadas como caixas retangulares e os relacionamentos são estabelecidos vinculando essas caixas. Cada estilo usa uma notação de relacionamento diferente, e um conjunto de símbolos distintos para representar a cardinalidade entre as tabelas conectadas, e cada um usa maneiras diferentes de representar as colunas.

Uma comparação dessas quatro notações é ilustrada na Figura 2-1, uma adaptação de Carlos Coronel e Morris (2019), contendo uma lista dos principais símbolos utilizadas no modelo ER, a nomenclatura utilizada e a descrição de cada símbolo.

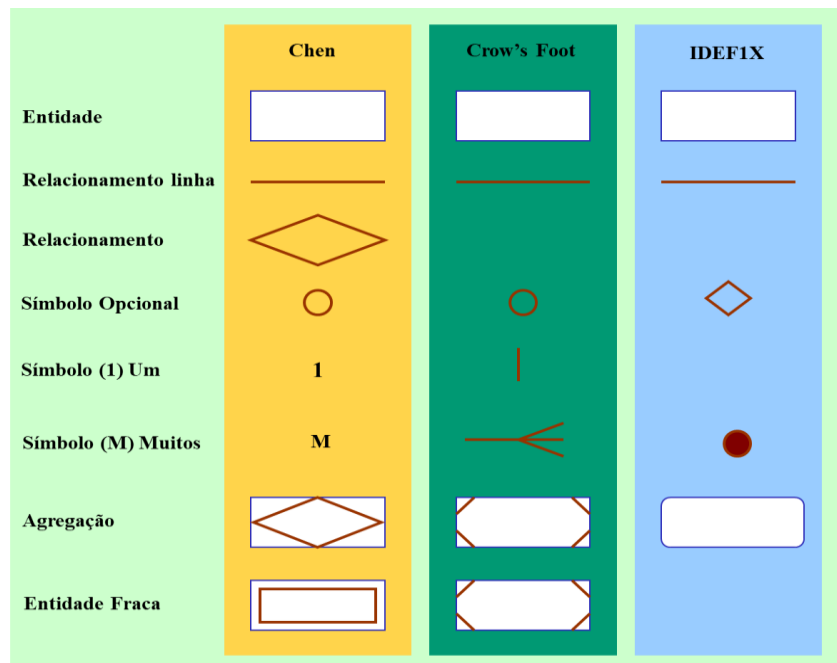


Figura 2-1. Comparação das notações de modelo ER, adaptado de (Coronel & Morris, 2019).

As notações para construção de diagramas são amplamente aceitas pela comunidade científica e acadêmica, e também são muito utilizadas pelas ferramentas de modelação. A notação *crow's foot* também é conhecida como engenharia da informação, seja em ferramentas CASE como no caso do ERwin, ou em outros livros didáticos, por exemplo de (Kroenke, Auer, Vandenberg, & Yoder, 2018). As subsecções a seguir descrevem as principais características das notações de Chen, IDEF1X e *crow's foot*.

2.2.1.1 Notação Chen

O diagrama de entidade e relacionamento foi introduzido por Chen em 1976 (P. P.-S. Chen & Pin-Shan, 1976). O modelo de Chen promoveu o modelo conceptual para um cenário prático de projeto de BD e estabeleceu os componentes básicos de construção: entidades e relacionamentos.

A notação original de Chen representava apenas entidades, relacionamentos, atributos e cardinalidade (Saiedian, 1997). Essa notação foi estendida por Teorey, Yang e Fry (1986), e desde então, o modelo E-R estendido de Chen é o mais amplamente utilizado (Teorey, Yang, & Fry, 1986).

Saiedian (1997) relatou as modificações que a notação de Chen recebeu ao longo de duas décadas, sendo as alterações mais relevantes: os atributos compostos, os identificadores, cardinalidade, participação, agregação, generalização e especialização.

A Figura 2-2 ilustra um exemplo de DER na notação Chen, onde as entidades são representadas por uma caixa e os tipos de relacionamento são simbolizados por um losango. Um retângulo duplo e um diamante duplo representam um tipo de entidade fraco e um relacionamento fraco, respetivamente. Atributos são representados por símbolos ovais e quando exibidos, aparecem em elipses (círculos alongados) conectados com uma linha à entidade ou relacionamento ao qual pertencem. A cardinalidade máxima de cada relação possui o símbolo 1 para “um” ou M para “muitos”, a cardinalidade mínima não é apresentada.

O modelo da Figura 2-2 pode ser lido da seguinte forma:

- cada CLIENTE *pode gerar* uma ou mais FATURAS;
- cada FATURA *é gerada* por um CLIENTE;
- cada FATURA *contém uma ou mais linhas* de fatura;
- cada LINHA da fatura está contida numa FATURA;
- cada FATURA da fatura faz referência a um PRODUTO;

- cada PRODUTO pode ser referenciado em uma ou mais linhas de FATURA;

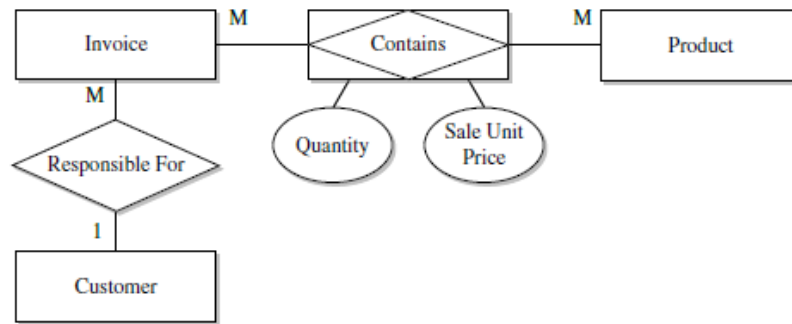


Figura 2-2. Exemplo de diagrama na notação CHEN, extraído de (Oppel, 2011).

Elmasri e Navathe (2015) também contribuíram para desenvolver o modelo Chen, e acabaram por criar uma notação própria do DER, que é uma variação do Chen original, que pode ser vista no livro de sistemas de bases de dados (Elmasri & Navathe, 2015). Embora o modelo Chen e suas variações já não sejam a referência dominante em DER, todas as ferramentas CASE atuais encontram suas origens conceptuais no modelo Chen (Coronel & Morris, 2019).

2.2.1.2 Notação IDEF1X

O IDEF1X é um modelo binário que não permite relacionamentos n-ários ou relacionamentos muitos-para-muitos com atributos não-chave (Bruce & A., 1992). A Figura 2-3 ilustra um exemplo de DER na notação IDEF1X.

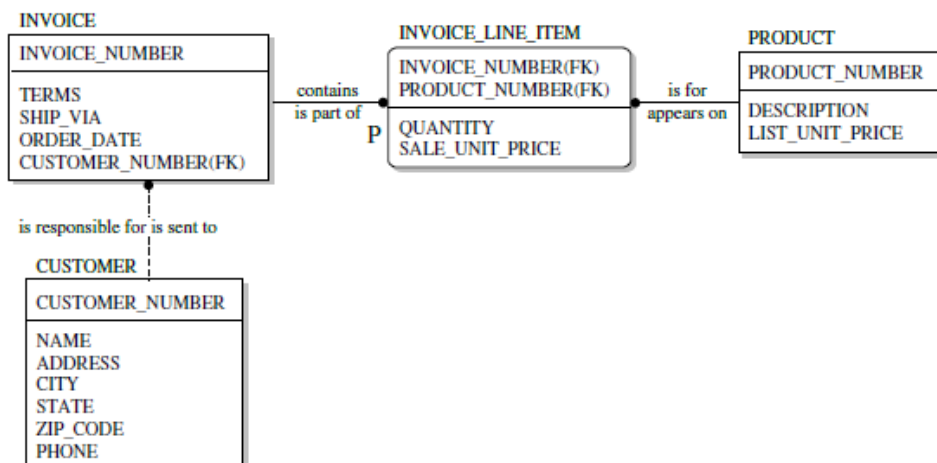


Figura 2-3. Exemplo de diagrama na notação IDEF1X, extraído de (Oppel, 2011).

Na notação IDEF1X, qualquer objeto com pelo menos um atributo portador de informações é modelado como um tipo de entidade. O tipo de entidade regular é chamado de entidade independente e é representado por uma caixa fechada com o nome da entidade no topo. Os atributos da entidade são listados dentro da caixa. As chaves primárias estão listadas

na seção superior da caixa. Os atributos de dados são anotados na seção inferior da caixa. Entidades independentes ou entidades pai são entidades que não dependem de outra entidade para sua identificação. E, a maioria dos relacionamentos é um-para-um, um-para-muitos ou muitos-para-muitos sem atributos não-chave.

O IDEF1X usa menos símbolos do que os outros métodos de modelação, fornecendo assim menos detalhes explícitos do tipo e da extensão dos relacionamentos que estão sendo modelados.

2.2.1.3 Notação engenharia da informação ou *crow's foot*

A notação de engenharia da informação ou *crow's foot* foi concebida e aprimorada entre as décadas de 60 à 80, de entre os diversos trabalhos tem-se a contribuição de Bachman (1969), que identificou como as entidades podem ser visualmente representadas através de caixas retangulares e também como representar as associações entre as entidades por setas (Bachman, 1969). Outra contribuição está centrada na especificação dos tipos de relacionamentos, “um-para-um”, “um-para-muitos”, e “muitos-para-muitos” denotado pelos seguintes símbolos (1:1, 1:N, M:N), foram concebidos por Gordon Everest (1976), que originalmente denominou essa notação como “seta invertida” em contraste com a notação de Bachman (Gordon C. Everest, 1976). O nome *crow's foot* é um reflexo do símbolo usado para a conectividade M, que se assemelha ao pé de uma ave de três pontas. O modelo combina informações de conectividade e cardinalidade num único conjunto de símbolos.

Clive Finkelstein (1981) descreveu o processo de modelação numa série de seis artigos publicados pela US Computerworld descritos com o mesmo nome “engenharia da informação” (Finkelstein, 1998). A identificação visual do tipo dos relacionamentos proposta por Clive Finkelstein é baseada na notação *crow's foot*, com adaptações para incluir os símbolos de mínimos e máximos da participação. Ainda, essa notação inclui separadamente um dicionário de dados, que lista as entidades, os atributos e uma descrição usada para a formação de procedimentos. Os artigos quatro e cinco da série publicada na Computerworld exemplificam com ilustrações (ver Figura 2-4) o processo de modelação usando a notação “Engenharia da Informação” (Finkelstein, 1981a, 1981b).

No mesmo ano de 1981 um livro foi publicado por Clive Finkelstein com coautoria de James Martin, onde descreveram com mais detalhes o processo de modelação usando a notação “engenharia da informação” (Martin & Finkelstein, 1981). Ambos os autores continuaram a contribuir de formas distintas na melhoria do processo de modelação. Finkelstein evoluiu com a variante orientada para negócios, e Martin evoluiu para a variante orientada pelo

processamento de dados. O então diretor técnico, Charles M. Richter, de 1983 a 1987, orientado por Clive Finkelstein, desempenhou um papel significativo, reformulando a metodologia e ajudando a projetar o produto de software que ajudou a automatizar o processo de modelação (Finkelstein, 1998).

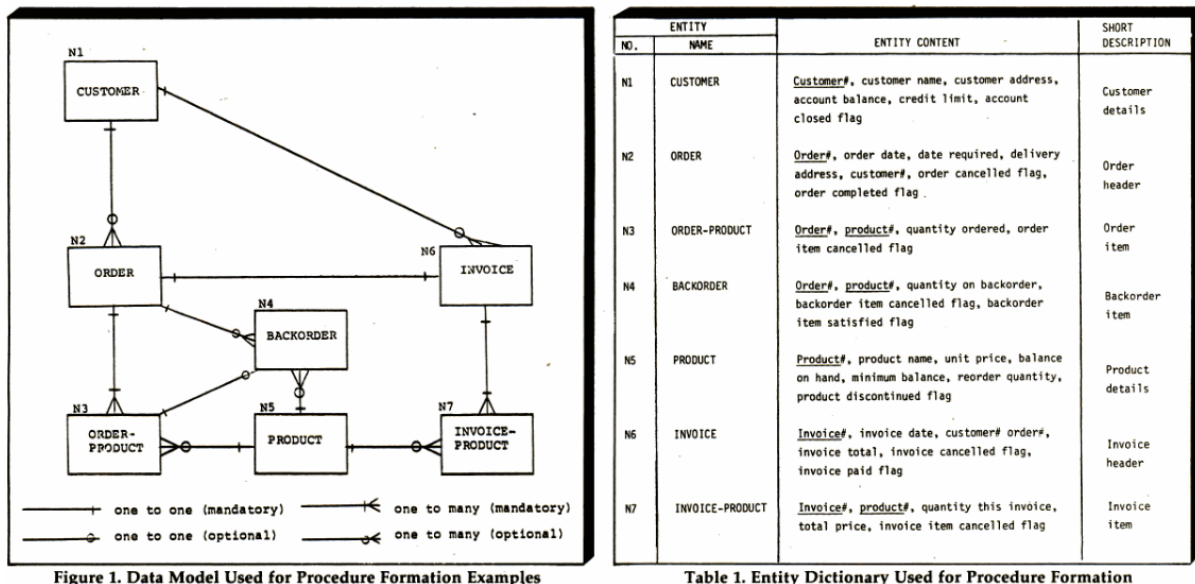


Figura 2-4. Exemplo do DER na notação engenharia de informação, extraída da revista nº 5 publicada na Computerworld (Finkelstein, 1981b).

O encadeamento Martin foi orientado pelo projeto de bases de dados desde o início e, desde 1983, focou-se na possibilidade de automatizar o processo de desenvolvimento através do fornecimento de técnicas para descrição de negócios que poderiam ser usadas para preencher um dicionário ou enciclopédia de dados que, por sua vez, poderia ser usado como material de origem para gerar código. A metodologia Martin forneceu uma base para a indústria de ferramentas CASE (engenharia de software auxiliada por computador). O próprio Martin teve participações significativas em pelo menos quatro fornecedores de ferramentas CASE - *InTech (Excelerator)*, *Higher Order Software*, *KnowledgeWare*, originalmente *Database Design Inc*, *Information Engineering Workbench* e *James Martin Associates*, originalmente *DMW* e agora *Headstrong* (os projetistas originais da *Texas Instruments CA Gen* e os principais desenvolvedores da metodologia) (Finkelstein, 1998).

No final dos anos 80 e início dos anos 90, o segmento Martin incorporou o desenvolvimento de aplicações rápidas (RAD) e a reengenharia de processos de negócio (BPR) e logo depois também entrou no campo orientado a objetos. Nesse mesmo período, o segmento de Finkelstein evoluiu ainda mais para a *Enterprise Architecture (EA)* e seus métodos de ITE direcionados aos negócios evoluíram para a *Enterprise Engineering* para a rápida entrega da

EA. Isso é descrito no livro “Arquitetura corporativa para integração: métodos e tecnologias de entrega rápida” primeira edição por (Finkelstein, 2006).

Atualmente, essa notação é conhecida pelo nome do artigo original “Engenharia da Informação”, também com o nome do autor James Martin, e ainda com as variações do apelido *crow’s foot*. No livro mais recente de BD do Kroenk et al. (2019), essa notação é denominada “*IE Crow’s foot Model*”, e no livro de Carlos Coronel e Morris (2019) é denominado “*Crow’s foot*”. Independente da variação do nome, essa notação nos livros está fortemente ligada à indústria. Ou seja, fornecem um conjunto muito rico de conceitos de modelação com muitos exemplos e ilustrações usando as ferramentas mais amplamente utilizadas de modelação. Ambos os livros oferecem detalhes para modelar usando ferramentas CASE, como ERwin, Microsoft Visio, e adicionalmente mostram como implementar o modelo usando os SGBD, destacando as diferenças entre as notações existentes fornecidas pelos SGBD como Microsoft Access, SQL Server, Oracle, MySQL e Postgres (Coronel & Morris, 2019; Kroenke et al., 2018).

Considerando que existem variações da nomenclatura da notação “*IE Crow’s foot*” e diferenças visuais dessa notação usadas nas ferramentas CASE e nos SGBD, padronizámos a escrita da tese com a nomenclatura *crow’s foot*. Bem como adotámos como referencial teórico o livro de (Coronel & Morris, 2019).

A notação *crow’s foot* também é um método binário que não permite um relacionamento ternário e não apresenta atributos relacionados com um relacionamento. As restrições de cardinalidade e participação são combinadas em notação min/max (barra e pé). A Figura 2-5 ilustra o exemplo DER usando a notação “*IE Crow’s foot*”. O tipo de entidade é representado como uma caixa. O relacionamento apresenta uma linha que liga duas entidades associadas e dando um nome, semelhante à notação proposta por Clive Finkelstein

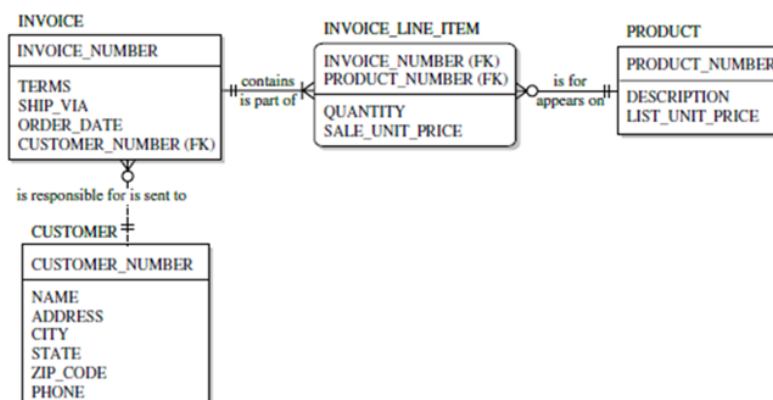


Figura 2-5. Exemplo de diagrama na notação *crow’s foot*, extraído de (Oppel, 2011).

Ao contrário da metodologia de Chen, a notação *crow's foot* não pode detalhar cardinalidades diferentes de 0, 1 ou N. Por exemplo, a cardinalidade (5,25) não pode ser apresentada num DER na notação *crow's foot*. No entanto, existem ferramentas de modelação comercial que permitem adicionar essas cardinalidades ao diagrama, como o Microsoft Visio.

2.2.2 Métodos de Avaliação de DER

O DER pode ser considerado um produto de software que pode ser avaliado na perspectiva de qualidade dentro de um processo de engenharia de software, bem como, pode ser um objeto de aprendizagem, e ser avaliado num processo de ensino de BD.

Uma série de estruturas de qualidade tem sido proposta na literatura para avaliar o DER, mas nenhuma delas foi amplamente aceite na prática e nenhuma emergiu como um padrão em potencial (D. L. Moody, 2005). Moody (2005) chegou a esta constatação após rever mais de 60 propostas sobre avaliação do modelo de dados conceptual, e descobriu que menos de 20% dessas propostas foram empiricamente validadas, e que a maioria são orientadas para a deteção da qualidade do produto de software. Segundo Moody (2005), nenhuma das soluções propostas é aceite na prática, fora do ambiente de investigação. As soluções propostas mostram falta de concordância com a terminologia, falta de consistência com padrões e campos relacionados, falta de procedimentos de avaliação e métricas, falta de diretrizes para melhoria (as soluções propostas concentram-se principalmente na deteção de erros), falta de atenção à qualidade do processo (ou seja, processo de criação de modelos conceptuais de dados e prevenção de erros), e falta de estudos empíricos na prática (ou seja, estudos sobre como a avaliação conceptual do modelo de dados é feita na prática) (D. L. Moody, 2005).

Um padrão internacional, ISO/IEC 25012:2008, foi estabelecido para avaliar a qualidade dos produtos de software. Este padrão pode ser usado para estabelecer requisitos de qualidade de dados, definir medidas de qualidade de dados ou planear e executar avaliações de qualidade de dados (ISO/IEC, 2008). Alguns problemas anteriormente levantados como falta de concordância com a terminologia, consistência, são solucionados pelo padrão ISO/IEC 25012:2008, no qual a qualidade dos dados é definida como “o grau em que as características dos dados atendem às necessidades declaradas e implícitas quando usadas em condições especificadas” e fornece “um modelo geral de qualidade dos dados retidos num formato estruturado num sistema de computador” (ISO/IEC, 2008). Entretanto, o padrão não fornece métodos ou procedimentos objetivos para avaliar os DER, como por exemplo uma métrica.

Uma investigação mais recente fornece um estado da arte na área de qualidade dos dados e informações através da descrição sistemática e comparativa do grande número de investigações relacionadas com a qualidade dos dados (Batini & Scannapieco, 2016). Batini e Scannapieco (2016) demonstram a limitação do padrão ISO/IEC 25012:2008 para se medir as estruturas dos dados (ver Tabela 2-4), “as dimensões de dados e dimensões de esquema são geralmente definidas de forma qualitativa, referindo-se a propriedades gerais dos dados e a esquemas de BD (DER físico), e as respetivas definições não fornecem qualquer mecanismo para atribuir valores às próprias dimensões” (Batini & Scannapieco, 2016).

Tabela 2-4. Dimensões de qualidade de dados padrão ISO/IEC 25012:2008, extraído de (Batini & Scannapieco, 2016).

Dimensão de qualidade do dado	Definição (todas as definições, exceto integridade e acessibilidade, começam com: o grau em que os dados têm atributos que ...) ...
Correção	representar corretamente o verdadeiro valor do atributo pretendido de um conceito ou evento num contexto específico de uso.
Compleitude	dados do assunto associados a uma entidade possuem valores para todos os atributos esperados e instâncias de entidades relacionadas num contexto específico de uso.
Consistência	estão livres de contradições e são coerentes com outros dados num contexto específico de uso.
Credibilidade	são considerados verdadeiros e confiáveis pelos utilizadores num contexto específico de uso.
Atualidade	têm a idade certa num contexto específico de uso.
Acessibilidade	os dados podem ser acedidos num contexto específico de uso, principalmente por pessoas que precisam de tecnologia de suporte ou configuração especial devido a alguma deficiência
Conformidade	aderir às normas, convenções ou regulamentos em vigor e regras semelhantes relacionadas à qualidade dos dados num contexto específico de uso
Confidencialidade	garantir que seja acessível e interpretável apenas por utilizadores autorizados num contexto específico de uso.
Eficiência	pode ser processado e fornecer os níveis esperados de desempenho usando as quantidades e os tipos apropriados de recursos num contexto específico de uso
Precisão	são exatas ou que discriminam num contexto específico de uso
Rastreabilidade	fornecer um registo de auditoria de acesso aos dados e de quaisquer alterações feitas nos dados num contexto específico de uso.
Compreensibilidade	permitem que seja lido e interpretado pelos utilizadores e são expressos em idiomas, símbolos e unidades apropriados num contexto específico de uso.
Disponibilidade	permitem que seja lido e interpretado pelos utilizadores e são expressos em idiomas, símbolos e unidades apropriados num contexto específico de uso.
Portabilidade	permita que seja instalado, substituído ou movido de um sistema para outro, preservando a qualidade existente num contexto específico de uso.
Recuperabilidade	permita que ele mantenha e preserve um nível especificado de operações e qualidade, mesmo em caso de falha, num contexto específico de uso.

A partir de uma série de limitações encontradas no padrão ISO/IEC 25012:2008, Batini e Scannapieco (2016) estabelecem um *framework* para a qualidade dos dados e informação e utiliza métricas de engenharia de software para estimar as dimensões de qualidade do dado, que pode ser utilizado inclusive nas múltiplas dimensões tipicamente representadas nas bases de dados relacionais, “as definições não fornecem medidas quantitativas e uma ou mais métricas devem ser associadas às dimensões como propriedades distintas e separadas. Para cada métrica, um ou mais métodos de medição devem ser fornecidos com relação a (1) onde é realizada a medição, (2) quais os dados incluídos, (3) o dispositivo de medição e (4) a escala

em que resultados são relatados (Batini & Scannapieco, 2016). O *framework* proposto possui oito dimensões de qualidade, e as definições seguem na Tabela 2-5:

Tabela 2-5. Uma estrutura de classificação para dimensões de qualidade de dados e informações, extraído de (Batini & Scannapieco, 2016).

1. <i>Exatidão</i> , correção, validade e precisão concentram-se na adesão a uma determinada realidade de interesse.
2. <i>Completeness</i> , pertinência, relevância e integridade referem-se à capacidade de representar todos e somente os aspectos relevantes da realidade de interesse.
3. <i>Redundância</i> , minimalidade, compacidade e concisão se referem à capacidade de representar os aspectos da realidade de interesse com o uso mínimo de recursos informativos.
4. <i>Legibilidade</i> , compreensibilidade, clareza e simplicidade referem-se à facilidade de entendimento e fruição das informações pelos utilizadores.
5. <i>Acessibilidade</i> e disponibilidade estão relacionadas à capacidade do utilizador de aceder a informações da sua cultura, status / funções físicas e tecnologias disponíveis.
6. <i>Consistência</i> , coesão e coerência se referem à capacidade das informações de cumprir, sem contradições, todas as propriedades da realidade de interesse, conforme especificado em termos de restrições de integridade, edição de dados, regras de negócios e outros formalismos.
7. <i>Utilidade</i> , relacionada à vantagem que o utilizador obtém com o uso da informação.
8. <i>Confiança</i> , incluindo credibilidade, confiabilidade e reputação, capturando quanta informação deriva de uma fonte autorizada. O cluster de confiança também abrange problemas relacionados à segurança.

O *framework* proposto por Batini (2016) oferece uma estrutura para o desenvolvimento de novas soluções operadas por computador (ver Tabela 2-5), entretanto, apresentamos a seguir apenas as dimensões precisão, completude, e redundância, por terem mais utilidade neste projeto de investigação.

Os diagramas ER podem ser avaliados nas oito dimensões propostas no *framework* por Batini (2016), utilizando um conjunto de métricas fornecidas para cada dimensão, mas nem todas as dimensões são interessantes para o contexto educacional. Por isso, são apresentadas apenas três dimensões de qualidade, precisão, completude, e redundância que são mais apropriadas no contexto deste projeto.

Precisão: Existem dois tipos de precisão estrutural: uma precisão sintática e uma precisão semântica. A precisão sintática é a proximidade de um valor v aos elementos do domínio de definição correspondente D , são medidas por funções de comparação, que avaliam a distância entre v e os valores em D . A precisão semântica é a proximidade do valor v ao valor verdadeiro v_0 , a precisão semântica é melhor medida com um domínio <sim, não> ou <correto, não correto>. Para medir a precisão semântica de um valor v , o valor verdadeiro correspondente deve ser conhecido ou, caso contrário, deve ser possível, considerando o conhecimento adicional, deduzir se o valor v é ou não é o verdadeiro valor. Para precisão de relação e base de dados, para ambos precisão sintática e semântica, normalmente é calculada uma proporção entre valores precisos e o número total de valores (Batini & Scannapieco, 2016).

Compleitude: Segundo Batini e Scannapieco (2016), pode ser definida como “a medida em que os dados são suficientemente abrangentes, profundos e precisos para a tarefa em questão”. Existem três tipos de completude ou integridade: a) a integridade do esquema ou modelo físico é definida como o grau em que os conceitos e suas propriedades não estão ausentes no esquema; b) a integridade da coluna é definida como uma medida dos valores ausentes para uma propriedade específica ou coluna numa tabela; e c) a completude da população avalia os valores ausentes em relação a uma população de referência. Dependendo da profundidade da avaliação, é possível medir a ausência de qualquer propriedade do modelo, por exemplo, de relações, dividindo o total de relações pelo tamanho de relações ausentes, dado pela fórmula a seguir (Batini & Scannapieco, 2016).

$$C(r) = \frac{|r|}{|ref(r)|}$$

Redundância: Um esquema é mínimo se todas as partes dos requisitos forem representadas apenas uma vez no esquema. Ou seja, não é possível eliminar um elemento do esquema sem comprometer o conteúdo da informação. No modelo relacional, a normalização está estritamente relacionada com a estrutura das dependências funcionais. Vários graus de normalização têm sido definidos no modelo relacional, como primeira, segunda, terceira, quarta Boyce Codd, e outras formas normais. A forma normal mais popular e intuitiva é a forma normal de Boyce Codd (BCNF). Um esquema relacional R está na BCNF se para cada dependência funcional não trivial $X \rightarrow Y$ definida em R , X contém uma chave K de R , ou seja, X é uma superchave de R . Para exemplificar, um esquema relacional R está na BCNF se todas as dependências funcionais não triviais tiverem uma chave no lado esquerdo da dependência, então, todos os atributos não-chave dependem de uma chave única. A interpretação desta propriedade é que o esquema relacional representa um conceito único, com o qual todas as dependências funcionais não triviais estão homogeneamente associadas e cujas propriedades são representadas. Desta forma, um esquema desnormalizado apresenta um conjunto de anomalias em relação a um esquema normalizado correspondente. Isto é medido pela concisão, refere-se à presença de elementos irrelevantes em relação ao domínio ou à minimização de esquemas e elementos de dados redundantes. A concisão é classificada em nível do esquema e em nível de dados. A concisão, ao nível do esquema, mede o número de elementos de esquema únicos (ou seja, propriedades e classes) de um conjunto de dados em relação ao número total de elementos de esquema, num esquema. A concisão ao nível dos dados mede o número de entidades únicas em relação ao número total de entidades no conjunto de dados. Adicionalmente, a concisão ao nível dos dados pode ser medida como o número total de

instâncias que violam a regra de exclusividade em relação ao número total de instâncias relevantes (Batini & Scannapieco, 2016).

Investigações mais recentes no campo da automação da avaliação de modelos conceituais de dados também consideram o modelo conceptual como um produto de software (Bai, Meredith, & Burstein, 2018; Kazi, Radulović, Berković, & Kazi, 2017). Ou seja, também usam o *framework* proposto por Batini e Scannapieco (2016) e definem um conjunto de métricas para alcançar um objetivo específico de automação. No passado, no mesmo contexto do modelo ER, algumas ferramentas de software desenvolvidas como protótipos permitiam: a) análise da qualidade dos elementos do modelo conceptual de dados, com base na ontologia de domínio (Sugumaran & Storey, 2006), b) comparação do modelo conceptual de dados criado com outros modelos (D. L. Moody & Shanks, 2003), e c) raciocínio automatizado sobre a qualidade dos modelos conceituais de dados (Choppella, Sengupta, Robertson, & Johnson, 2007).

As métricas têm vindo a ser utilizadas para avaliar o modelo de dados conceituais, e podem ser classificadas como (Kazi et al., 2017):

- Quantitativo: verificação do número de entidades, relacionamentos e atributos com certas características de complexidade de elementos e modelo (Genero, Poels, & Piattini, 2008; Tavana, Joglekar, & Redmond, 2007)
- Qualitativo: julgamento subjetivo sobre características de qualidade como: completude, integridade, flexibilidade, abrangência, correção, simplicidade, integração, capacidade de implementação e precisão, consistência, confiabilidade, pontualidade, e exclusividade (D. Moody, 2003; Piprani & Ernst, 2008);
- Ontológicos: baseado na estrutura (adequação, estabilidade, consistência) e baseado no conteúdo (integridade, coesão, validade) (Piprani & Ernst, 2008);
- Comportamental: características de aplicabilidade do projetista e utilizador, manutenção, correção e desempenho (Piprani & Ernst, 2008).

As métricas quantitativas podem fornecer um conjunto de medidas estruturantes do DER (Genero et al., 2008) e, dependendo do objetivo de automação do sistema, associado com as dimensões de qualidade dos dados.

Outra característica presente nessas investigações são a participação de especialistas e iniciantes na área de modelação, que tem o papel de avaliar empiricamente os DER, e dessa

forma são feitos progressos em direção à generalidade e aplicabilidade da estrutura de avaliação de modelo conceptual de dados proposta na prática (D. L. Moody & Shanks, 2003). Ainda, a verificação empírica da estrutura proposta é subjetiva no ranking das métricas dos critérios de qualidade, ou seja, o ranking dos modelos conceptuais de dados criados é realizado por pessoas qualificadas, habitualmente numa escala de 0 a 5 (0-fraco a 5-excelente). Existem diversas investigações relacionados com a automação na avaliação do modelo conceptual de dados (Boris, Nenad, & Svetlozar, 2013; Cagiltay, Tokdemir, Kilic, & Topalli, 2013; Kesh, 1995; D. L. Moody, 2005). Essas investigações resumem os diferentes métodos empregues na avaliação do DER, que se concentram no emprego de uma variedade de métricas para extrair as informações necessárias a serem aplicadas num contexto específico de automatização.

2.2.3 Desafios na Avaliação do DER

Existem diversos desafios que a avaliação do DER oferece. A seguir descrevemos as principais encontradas na literatura.

2.2.3.1 Notação do DER

As variações da notação do DER são o primeiro obstáculo a ser superado num processo de avaliação. Dependendo da notação é possível representar um mesmo conceito de formas diferentes, implicando em diferentes formas de implementação física na BD. Esse problema pode ser visto em (I.-Y. Song, Evans, & Park, 1995), que comparou pelo menos 10 notações de DER

Na perspetiva educacional, uma investigação avaliou a eficácia do ensino de várias notações DER, e concluiu que não há diferenças significativas entre as diferentes notações para entender o modelo, bem como para perceber a qualidade do diagrama, nem no nível de entendimento dos conceitos de ER (Pons, Polak, & Stutz, 2005). Porém, um outro estudo de investigação revelou que os alunos tendem a aprender melhor com DER que incluem mais especificações do nível lógico na notação, como por exemplo as especificações de chave primária e chave estrangeira. (Kamišalić, Heričko, Welzer, & Turkanović, 2018). Se por um lado há aqueles que preferem notações mais detalhadas, existem também aqueles que preferem com menos detalhes. Por exemplo, na investigação de (Puja, Posic, & Jaksic, 2019) uma avaliação subjetiva foi utilizada para classificar quatro notações de DER, numa escala cumulativa de 0 a 5 de quatro dimensões: velocidade de desenho, intuição, simplicidade e suporte. No final, Puja, Posic e Jaksic (2019) classificaram as notações de DER na seguinte ordem (IDEF1X com 19, UML com 17, Chen com 15, *crow's foot* com 13, e MIRIS com 12).

Ainda na perspectiva da educação, existem evidências de que as ferramentas visuais como MySQL Workbench auxiliam os alunos a aprender os conceitos de BD, uma vez que utiliza a notação *crow's foot* para desenhar o DER, ao mesmo tempo que manipulam fisicamente a BD (L. Yang & Cao, 2016). Conforme Yang e Cao (2016), os alunos aprendem mais rapidamente as relações 1:1, 1:M, e M:N fazendo o mapeamento visual das relações do DER nas ferramentas.

Por outro lado, na perspectiva empresarial, uma investigação experimental foi conduzida para descobrir a preferência das pessoas para documentar os requisitos de engenharia de software (Ibriwesh, Ho, Chai, & Tan, 2017). O ensaio revelou que os participantes preferem documentar os requisitos usando a perspectiva do DER, pois consideram mais clara, mais fácil e mais compreensível do que as outras duas perspectivas (linguagem natural e diagrama de classes). Também se descobriu, que os participantes do grupo de linguagem natural enfrentaram mais dificuldades em responder aos exercícios do que os dos grupos de DER e diagrama de classes. Ainda, os participantes da perspectiva DER gastaram menos tempo do que as duas outras perspectivas. Segundo Ibriwesh et al. (2017), esses resultados indicam que o uso de DER no domínio de aplicativos do mercado eletrônico é mais eficaz e útil na documentação de requisitos, do que as outras perspectivas.

As notações *crow's foot* e IDEF1X normalmente estão disponíveis nas ferramentas CASE, como no caso do ERwin, e são usualmente citadas nos principais livros de BD. Após o levantamento bibliográfico de cerca de 40 livros da área de BD (ver Anexo A: Lista de livros de BD), encontram-se factos históricos da evolução dessas notações e o momento em que o DER se tornou interessante para a indústria. Historicamente, sabe-se que apenas duas notações, *crow's foot* e IDEF1X tiveram uma *task force*, ou seja, uma equipa de pessoas especificamente apoiadas pelo governo e indústria que efetivamente procuraram evoluir a modelação com o objetivo de automatizar o DER para implementar fisicamente a BD (Finkelstein, 1998). Esse facto poderia ser levado em consideração para escolher ao menos uma notação para avaliar o DER. Mas sabe-se que, na prática, as variações da notação de Chen continuam a ser utilizadas, como no caso do livro de Elmasri e Navathe (2015), atualmente na sétima edição.

Seja na indústria ou na academia, encontramos uma variedade de notações que podem ser utilizadas para modelar a BD. Atualmente, existem diversas ferramentas online para edição de DER, e algumas delas são fornecidas gratuitamente. Isso aumenta o leque de opções, tanto na indústria como na academia a escolha da notação pode ser determinada por um gosto pessoal do utilizador ou por um outro fator, por exemplo, tempo de produtividade do DER.

A notação é um desafio pois, dependendo do estilo, pode representar diferentes níveis de especificação da BD, alguns com mais ou menos detalhes. Ainda, dependendo da notação é possível definir várias formas de relacionar um conjunto de entidades. Portanto, seria muito mais complexo avaliar um DER completo, que tenha um estilo de notação que represente a BD fisicamente, do que avaliar um diagrama na notação com menos detalhes da BD. Por exemplo, um DER físico na notação *crow's foot* versus um DER na notação original de Chen.

2.2.3.2 Tipos de questões

Ao pensar nas entradas e saídas de um sistema educacional que pretende avaliar o DER, certamente se chegará às questões que os utilizadores deverão responder. No fim dos capítulos dos livros didáticos de BD, normalmente é fornecida uma série de questões ao professor, bem como material suplementar para aferir o conhecimento dos alunos. A priori, as questões são descritas em linguagem natural, entretanto não se sabe ao certo que outros formatos essas questões podem vir a assumir. Nesse sentido, uma vez que o projeto está inserido no contexto acadêmico, procurou-se descobrir os diferentes formatos em que as questões de DER têm sido apresentadas nos livros didáticos de BD. Procurou-se responder às seguintes questões: a) quais são os livros de BD que oferecem questões de DER? e b) quais são as diferentes formas de criar uma questão de DER?

A priori, as respostas a essas questões permitem limitar o propósito das aplicações na avaliação de DER, e auxiliar na elaboração de uma base de questões de DER que possam ser utilizadas na automatização. Para isso foi feita uma busca dos livros mais citados em cursos de BD no Google Acadêmico, além dos livros adotados por instituições de ensino superior e também indicados pela ACM e IEEE nos seus guias curriculares dos cursos de ciência e tecnologia informática. Foram encontrados 49 livros no domínio de BD e DER, variando de clássicos desde 1985 às publicações de livros mais recentes até ao ano de 2016.

No entanto, muitos desses livros foram descartados por não abordarem o tópico do DER, no todo ou em parte. A Figura 2-6 ilustra a quantidade de livros selecionados por ano com a legenda “Sim” na cor verde, e com a legenda “Não” na cor vermelha para os livros excluídos da seleção. Os critérios de exclusão dos livros foram a falta de diagramas ilustrativos para explicar os conceitos de ER, o uso de um único exemplo descritivo para todos os conceitos de ER, o uso exclusivo da linguagem SQL para exemplificar conceitos de ER e a não apresentação de nenhuma lista de exercícios com respostas ou apoio ao professor.

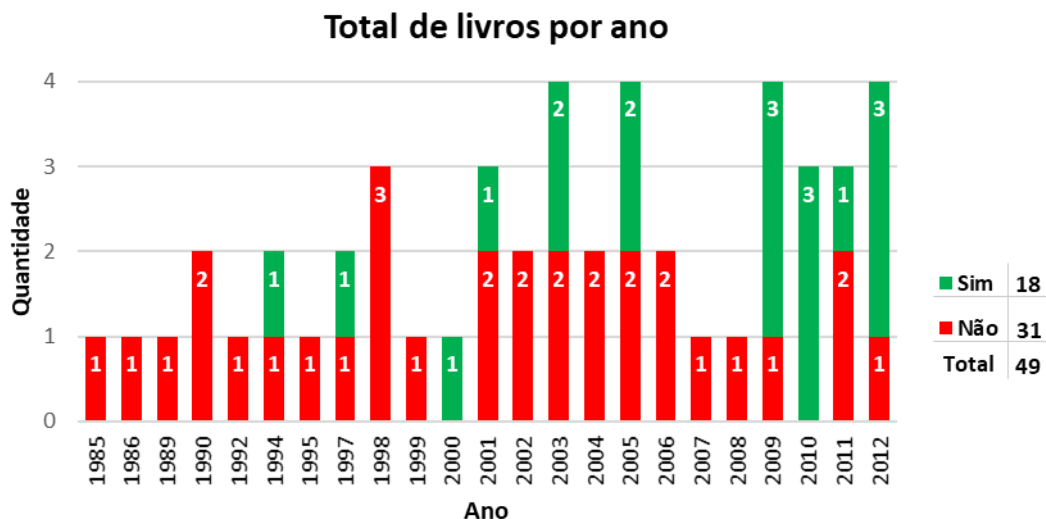


Figura 2-6. Quantidade de livros na área de BD com tópico de DER com exercícios e material de suporte para o professor (“Sim” em verde - livros selecionados, e “Não” em vermelho - livros rejeitados).

a) Quais são os livros de BD que oferecem questões de DER?

Ao todo, cerca de 18 livros da área de BD foram selecionados e oferecem uma lista de questões de DER com respostas para os professores. A lista dos livros com ISBN e ano que respondem essa primeira questão está disponível no Anexo A: Lista de livros de BD.

b) Quais são as diferentes formas de criar uma questão de DER?

Existem pelo menos três tipos básicos de apresentação de questões de DER, e foram denominadas como:

- (1) tipo de transcrição, são aquelas onde o aluno vê uma imagem para criar o DER;
- (2) o tipo de cenário em que o aluno lê uma descrição textual para criar o DER;
- (3) o tipo de normalização onde o aluno vê uma tabela não normalizada para criar o DER.

Esses três tipos de questões podem ser combinadas entre si, possibilitando assim uma maior diversificação para criar questões de DER. Por exemplo, poderia ser criado uma questão baseada nos tipos cenário e transcrição, ou cenário e normalização, ou transcrição e normalização, e assim por diante. A Tabela 2-6 exemplifica os tipos de questões de transcrição, cenário, e normalização com uma resposta do professor cadastrada no ambiente.

Tabela 2-6. Tipos genéricos de questões de DER.

#	Exemplo de Questão	Figura Associada	Um exemplo de resposta do professor																								
Transcrição	Crie o diagrama relacional para mostrar a relação entre o DIRECTOR e o PLAY. Veja a figura ao lado:	<p>Table name: STUDENT</p> <table border="1"> <thead> <tr> <th>STU_CODE</th> <th>PROF_CODE</th> </tr> </thead> <tbody> <tr><td>100278</td><td></td></tr> <tr><td>128569</td><td>2</td></tr> <tr><td>512272</td><td>4</td></tr> <tr><td>531235</td><td>2</td></tr> <tr><td>531268</td><td></td></tr> <tr><td>553427</td><td>1</td></tr> </tbody> </table> <p>Table name: PROFESSOR</p> <table border="1"> <thead> <tr> <th>PROF_CODE</th> <th>DEPT_CODE</th> </tr> </thead> <tbody> <tr><td>1</td><td>2</td></tr> <tr><td>2</td><td>6</td></tr> <tr><td>3</td><td>6</td></tr> <tr><td>4</td><td>4</td></tr> </tbody> </table>	STU_CODE	PROF_CODE	100278		128569	2	512272	4	531235	2	531268		553427	1	PROF_CODE	DEPT_CODE	1	2	2	6	3	6	4	4	
STU_CODE	PROF_CODE																										
100278																											
128569	2																										
512272	4																										
531235	2																										
531268																											
553427	1																										
PROF_CODE	DEPT_CODE																										
1	2																										
2	6																										
3	6																										
4	4																										
Transcrição	Crie o diagrama relacional para mostrar a relação entre o MÉDICO e PACIENTE. Veja a figura ao lado:																										
Cenário	Crie um diagrama relacional para armazenar informações sobre a produção artística. Observe, em particular, as seguintes regras de negócios que são refletidas nas tabelas e seu conteúdo: a) Um pintor pode pintar muitas pinturas, e todo pintor tem um código e nome. b) Cada pintura é pintada por apenas um pintor, e todas as pinturas têm um título.	Sem figura associada.																									
Normalização	Funcionários = {Id+Nome+Id_Cargo+Descricao_Cargo}	Sem figura associada.																									

Os DER no contexto da avaliação são muito ricos. Eles têm o intuito de investigar várias questões de avaliação. Uma ampla variedade de estilos de questões pode ser usada para avaliar uma solução básica de DER. Os problemas que exigem a produção de um DER, podem ter uma única solução correta, que é idêntica à solução ideal do professor, ou soluções alternativas equivalentes. Análogo a uma questão objetiva, as respostas dos alunos em diagrama têm a avaliação com um único DER correto, exigem um esquema de marcação e uma combinação de padrões simples com pouca interpretação (Thomas et al., 2005). Por outro lado, o processo de marcação de uma questão com o alternativo DER correto, pode exigir uma grande quantidade de conhecimento de domínio e um estilo diferente de esquema de avaliação com combinação de padrões complexos. Além disso, as soluções corretas dos problemas podem variar de pequenas a grandes ou de simples a complexas.

A conceção do modelo ER é uma tarefa complexa, envolve a identificação de factos relevantes de diferentes fontes de informação, muitas das quais são baseadas em texto. Batra e Davis argumentam que os alunos consideram a tarefa de MER difícil e exibem erros sistemáticos nos seus modelos (Dinesh Batra & Davis, 1992). Tais erros devem-se à falta de

compreensão e domínio do assunto e à falta de familiaridade com a tarefa. O aumento da exposição à modelação prática de ER melhoraria a capacidade dos alunos na projeção da BD.

2.2.3.3 Ambiguidade

Embora muitos investigadores tenham aceite o facto de que os conceitos teóricos do DER representam uma visão natural do mundo real (P. P.-S. Chen & Pin-Shan, 1976; Coronel & Morris, 2019; Date, 2012; Elmasri & Navathe, 2015; Kroenke, Auer, Vandenberg, & Yoder, 2019; Silberschatz, Korth, & Sudarshan, 2010; Teorey et al., 2011), essa crença é baseada principalmente no senso comum e na intuição e não em descobertas empíricas. Como bem apontado por Drazena e Ivica (2017), a popularidade do DER deve-se à “crença generalizada de que entidades e relacionamentos são conceitos naturais de modelação no mundo real” (Drazena & Ivica, 2017).

Ainda que o DER pareça bastante simples e direto nas primeiras definições, a literatura revela subtilezas inesperadas como em alguns conceitos de DER ambíguos. A construção da relação parece ser particularmente problemática no DER porque, muitas vezes, não se sabe ao certo quando usar relação, atributo ou mesmo entidade para representar algo no mundo real. Esses problemas foram exaustivamente discutidos, por exemplo, conforme Peckham e Maryanski (1988), “o conceito teórico de relação pode aparecer no modelo como um atributo, entidade, elemento independente ou função” (Peckham & Maryanski, 1988). Edgar Codd (1990) também afirmou que: “A distinção entre entidades e relacionamentos não foi, e ainda não é definida com precisão. Consequentemente, a entidade de uma pessoa é o relacionamento de outra pessoa.”. Goldstein e Storey (1990) descobriram que os utilizadores de um projeto de bases de dados automatizado tinham dificuldade em distinguir entre relacionamentos e atributos (Goldstein & Storey, 1989). Batra, Hoffler e Bostrom (1990) compararam as representações do DER e destacaram que “o erro mais notável encontrado nas soluções preparadas pelos sujeitos foi a representação incorreta da conectividade dos relacionamentos.” (D. Batra, Hoffler, & Bostrom, 1990). Noutro trabalho, Batra e Antony (1994) mostraram as dificuldades que os novatos enfrentaram em modelar entidades e relacionamentos (D. Batra & Antony, 1994). Chris Date (1995) também identificou o mesmo problema, afirmando: “A abordagem de ER é seriamente imperfeita porque o mesmo objeto pode ser legitimamente considerado como uma entidade por alguns utilizadores e como um relacionamento por outros.”. Da mesma forma, Elmasri e Navathe (2015) observam que “Às vezes, é conveniente pensar num tipo de relacionamento binário em termos de atributos” (Elmasri & Navathe, 2015).

Uma vez que alguns elementos do DER possuem mais de uma interpretação ou podem significar coisas diferentes, então pode admitir-se que são ambíguos. Desta forma, questiona-se então a naturalidade e a intuição dos constructos do DER.

2.2.3.4 Equivalência

O problema da equivalência dos modelos ER é outro desafio da avaliação do DER, ou seja, decidir quando dois DER agrupam a mesma informação (Barker, 1992; Jajodia, Ng, & Springsteel, 1983). Conforme Jajodia et al. (1983), dada uma descrição de requisitos de um sistema de BD, é possível obter vários DER, DER1, DER2, *, DERn estruturalmente diferentes. Cada um dos quais “está em conformidade” com a descrição da aplicação, num sentido informal.

No contexto educacional, dado um comando de um exercício de DER, os alunos também produzem várias respostas de DER diferentes. Veja a Figura 2-7, que ilustra três respostas de alunos para o mesmo problema de DER, onde se deseja armazenar as informações do grau de escolaridade do funcionário. Então, é natural e importante questionar se esses diferentes DER representam as mesmas informações sobre um determinado “mundo real”. Por exemplo, pode-se questionar: se (e em que sentido) os DER 1, 2 e 3 da Figura 2-7 são equivalentes? Ainda, dada uma escala qualquer, qual deveria ser a pontuação de cada resposta?

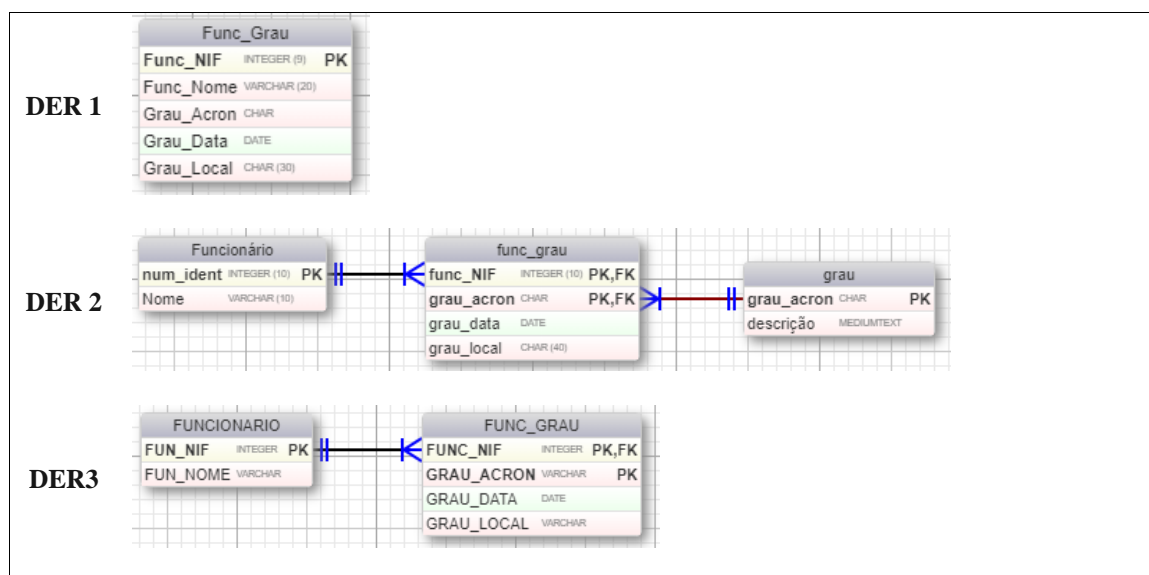


Figura 2-7. Exemplo de DER equivalentes para guardar o grau de escolaridade dos funcionários.

Como apontado por Jajodia et al. (1983), ainda que os DER sejam produzidos para representar o mesmo sistema, eles devem, de alguma forma, incorporar a mesma informação do “mundo real”. Entretanto, como os requisitos do sistema são geralmente fornecidos em linguagem natural, a tradução desses requisitos para os vários DER, por um processo de modelação, é baseada no conhecimento intuitivo do “significado” da descrição. Desta forma,

seria muito difícil formalizar uma noção de equivalência sob os requisitos do sistema, devido à imprecisão do próprio processo de modelação (Jajodia et al., 1983).

Assim, o nosso estudo concentra-se em encontrar a equivalência nos próprios DER a partir de um formalismo que identifica e pondera as informações semânticas que elas possuem. Portanto, se as nossas definições formais permitirem mostrar a equivalência entre os DER estruturalmente diferentes, então poderemos, de certa forma, confiar que esses DER são consistentes com os requisitos do sistema. Em seguida, um aluno num ambiente educacional, munido com esse conhecimento de equivalência de DER, poderá responder diversas vezes até encontrar uma solução mais equivalente a uma resposta do professor.

Em resumo, a ideia da equivalência consiste em medir e alcançar o “mesmo universo da informação”, ao invés de impor as mesmas estruturas nos DER. Assim, a definição trivial de equivalência pode ser dada como: dois DER são equivalentes se, e somente, forem idênticos, sem que as estruturas sejam necessariamente iguais!

A nossa proposta de equivalência entre DER centra-se em se ter a certeza de que os diagramas estão a tentar modelar o mesmo cenário do “mundo real”, e a avaliar as diferenças entre essas estruturas. Jajodia et al. (1983) estabeleceu os seguintes itens de equivalência de DER em relação aos domínios de dados:

- a) se a coleção total de conjuntos de entidades e valores descrevem suficientemente todos os objetos que estão a ser modelados;
- b) se a coleção total de conjuntos de relacionamentos define relações sobre a coleção total de conjuntos de entidades, e entidades relacionadas são mapeadas por atributos para conjuntos de valores;
- c) se a extensão dos domínios para os vários atributos (ou seja, seus conjuntos de valores) são os mesmos nos modelos que estão a ser considerados (Jajodia et al., 1983).

A nossa ideia de equivalência transcende os trabalhos anteriores, pois usa parte do formalismo não apenas para descobrir as semânticas dos DER, mas também pretende medir cada uma das singularidades das estruturas dos diagramas. Isso cria opções para responder à questão inicialmente elaborada sobre como devem ser avaliados, numa determinada escala, diferentes DER. A solução para este problema é dada no capítulo 3.

2.2.3.5 Principais problemas

Outras dificuldades relacionadas com a criação de modelos conceituais de dados foram documentadas em investigações anteriores (Antony & Batra, 2002; Dinesh Batra, 2007; Currim, 2008; Dey, Storey, & Barron, 1999; Liao & Palvia, 2000; D. L. Moody, 2004; Shoval & Shiran, 1997). Apesar de sua importância, há investigações que sugerem que a modelação conceptual de dados não é bem-sucedida, e por isso deve ser aprimorada, tanto em formação quanto em educação (Simsion, 2014). Investigadores estudaram o processo da modelação de dados conceptual utilizado por projetistas iniciantes visando perceber os fatores que motivam os erros (D. Batra & Antony, 1994; Currim, 2008; Simsion, 2014). Esses fatores são importantes na construção de ferramentas e técnicas que podem impedir os erros e melhorar a qualidade dos sistemas de informação. Os fatores são:

1. Ambiguidade

A partir da revisão da literatura foram identificados dois tipos de ambiguidade, a conceptual e na especificação de requisito. A ambiguidade conceptual ocorre quando os autores divergem sobre um mesmo conceito no DER. Por exemplo, Pendegraft (2012) identificou pelo menos dez autores que definem de forma ambígua a integridade referencial. Por outro lado, a ambiguidade na especificação de requisito está associada com as entradas do processo de conceção do DER que, na maioria das vezes, são fornecidas em linguagem natural. Estas entradas são inerentemente ambíguas. Por exemplo, se a descrição do problema estiver incompleta, a análise resultante com base nessa abordagem também estará incompleta. Ou seja, a análise de requisitos incompleta ou ambiguidade na descrição do problema resultará num DER incompleto (I. Y. Song et al., 2015).

2. Complexidade combinatória

Investigações anteriores mostram que projetistas iniciantes têm mais dificuldade em modelar relacionamentos do que entidades (Topi & Ramesh, 2002). Batra e Anthony (1994, 2007) examinam o desempenho do projetista na modelação de exercícios abertos e descobriram que os novatos não têm apenas dificuldade em modelar relacionamentos unário e ternário, mas também têm dificuldade em modelar quaisquer tipos de relacionamentos, incluindo relacionamentos binários (D. Batra & Antony, 1994; Dinesh Batra, 2007). Uma das razões advém do facto de que, dado um conjunto de entidades, existe potencialmente um número muito grande de relacionamentos possíveis. À medida que o número de entidades aumenta, o número de possíveis relacionamentos aumenta a uma taxa combinatória (ver Figura 2-8).

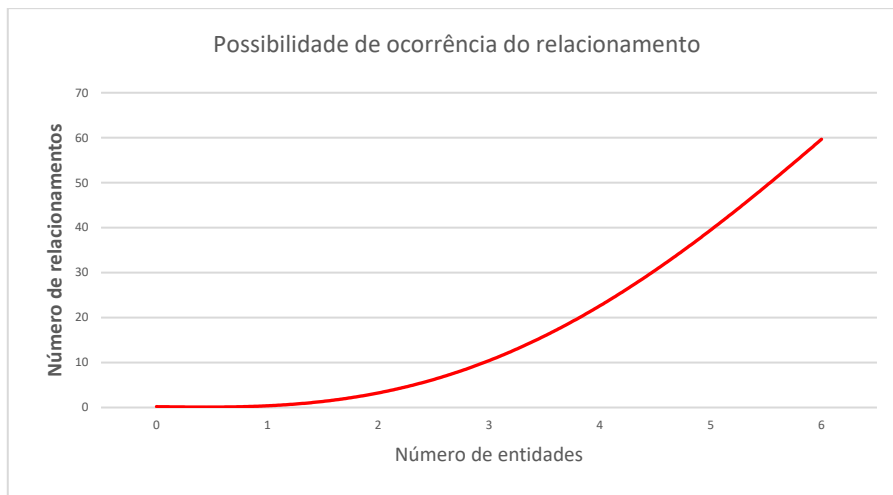


Figura 2-8. Número de possibilidades de relacionamentos por entidades.

Quais são as regras que permitem a um projetista escolher o conjunto certo de relacionamentos no modelo de ER? E como sabe o projetista se todos os relacionamentos corretos entre as entidades são identificados e quais estão incorretos? A maioria dos relacionamentos pode ser derivada de outros; portanto, o problema de modelação de relacionamentos é selecionar um conjunto mínimo que capture a semântica de forma eficaz e possa ser usado para derivar os outros. Para a identificação correta das construções de relacionamento, pelo menos os seguintes critérios devem ser atendidos: (1) nenhuma das semânticas na aplicação deve ser perdida, (2) nenhuma das construções de relacionamento deve ter relacionamentos redundantes, (3) e o grau de relacionamento deve ser mínimo.

3. Falha de correspondência semântica

Traduzir a especificação de requisitos literalmente em estruturas de bases de dados causa erros de tradução literais (Dinesh Batra, 2007). Por exemplo, uma frase que afirma “um pedido registra uma venda de produtos a clientes” pode incluir uma relação errada entre cliente e produto. Isto mostra que nem todas as relações do mundo real mapeiam relações de base de dados; algumas relações do mundo real são deriváveis ao nível de base de dados. Além disso, algumas relações do mundo real tornam-se indiretas, resultando em semântica ambígua. Relações indiretas sem relações diretas são erradas.

4. Inexperiência e conhecimento incompleto de projetistas novatos

Os projetistas iniciantes têm conhecimento e habilidades limitados, enquanto os especialistas geralmente usam suas experiências passadas. Mesmo um especialista pode não conseguir criar um modelo conceptual de qualidade devido à falta de conhecimento do domínio, a menos que tenha uma percepção clara das especificações de requisitos (Kim et al.,

2008). Portanto, é necessária experiência em conhecimento de domínio para identificar as entidades ocultas. As questões importantes são como os projetistas iniciantes podem ser formados com eficiência e como o conhecimento do domínio pode ser transferido para os projetistas.

5. Regras de modelação dispersas

Não existe um conjunto completo de heurísticas / regras que possam ser usadas no desenvolvimento de modelos de dados de qualidade (I. Y. Song et al., 2015). A partir das doze regras de Chen (1983) para construir modelos ER utilizando frases em inglês (P. P. S. Chen, 1983), vários estudos tentaram refinar e estender essa abordagem (Hartmann & Link, 2007; Lucassen et al., 2017). Essas regras são frequentemente úteis, mas às vezes podem levar a erros cognitivos (Parsons & Saunders, 2004). Essas regras ainda não são totalmente completas nem precisas, por exemplo, a regra de “Um substantivo comum corresponde a um tipo de entidade num diagrama de ER.”. Ainda que as entidades possam ser identificadas por substantivos numa especificação de requisitos, nem todos os substantivos são entidades, porque os substantivos não se referem apenas a entidades, mas também a atributos, ou outros conceitos descartáveis que não precisam ser modelados. Adicionalmente, uma entidade também pode ser identificada a partir de uma frase verbal e requisitos implícitos (I.-Y. Song et al., 2011). Essas regras são conflituosas e podem fornecer uma recomendação errada (I. Y. Song et al., 2015).

6. Múltiplas soluções corretas versus solução única

No projeto conceptual, não há uma resposta única nem um algoritmo para criar a melhor resposta. Moody e Shanks (1994) afirmam que um dos problemas comuns encontrados na modelação é o grande número de DER alternativos que podem ser criados para um problema específico (D. L. Moody & Shanks, 1994). Entretanto isto é um problema de DER equivalentes, e poderia ser evitado aceitando apenas uma única solução correta, neste caso, a solução DER pode estar certa ou errada. Por outro lado, aceitar múltiplas soluções corretas implica procurar soluções para a equivalência dos DER. Diferentes propostas podem ser investigadas neste caso, por exemplo Moody (1998) propõe uma estrutura de seis elementos para avaliar a qualidade dos modelos conceptuais de dados (D. L. Moody, 1998). As suas estruturas são compostas de integridade, simplicidade, flexibilidade, capacidade de compreensão, integração e capacidade de implementação. Posteriormente, foram acrescentados oito fatores nessa estrutura, incluindo correção e integridade, validando empiricamente a estrutura (D. L. Moody & Shanks, 2003).

Ferramentas de Avaliação Automática de DER

Este capítulo descreve o estado da arte das soluções de avaliação de DER (voltadas para ensino e aprendizagem) e realiza uma análise pormenorizada dos recursos disponibilizados aos alunos com o objetivo de encontrar lacunas que devem ser preenchidas na proposta de um novo modelo.

3.1 Considerações iniciais

Uma revisão da literatura foi conduzida com o objetivo de determinar quais eram as abordagens processadas por computador utilizadas para avaliar automaticamente os DER em ambientes educacionais. O protocolo estabelecido para revisão seguiu:

- Procura bibliográfica: foram utilizadas cinco bases de dados eletrônicas, ACM, IEEE, Springer, ScienceDirect e Web of Science. Os termos usados para artigos elegíveis foram “automatic evaluation, automatic feedback, automatic classification, electronic learning, virtual learning, virtual learning, network learning, diagram drawing, entity relationship modeling, ER modeling, automated diagram evaluation” e suas combinações. Os critérios de busca da data de publicação incluíram todos os trabalhos publicados desde 1976, quando foi desenvolvida a teoria DER (P. P.-S. Chen & Pin-Shan, 1976), até 3 de dezembro de 2017, quando a última busca foi realizada.
- Documentos elegíveis para seleção e estudo: o idioma inglês foi definido como critério de inclusão, considerando que esse idioma é favorecido pela comunidade científica na publicação de estudos de investigação. Os artigos considerados documentos elegíveis aplicaram abordagens processadas por computador para avaliar automaticamente DER em ambientes educacionais. Foram excluídos os

manuais, capítulos de livros, dissertações de mestrado e doutoramento, documentos não publicados, documentos em inglês e trabalhos exclusivos. Além disso, foram excluídos todos os trabalhos que não forneceram informações para análise dos resultados, aqueles que possuem uma proposta inovadora, mas não aplicaram a solução. Por mais sofisticadas que fossem as abordagens adotadas, esses trabalhos foram considerados não elegíveis, pois não permitiram a análise, identificação de limitações e validação de resultados.

- Os artigos que atenderam aos critérios de elegibilidade pré-especificados foram selecionados e os seus dados extraídos. A extração final dos dados do estudo incluiu detalhes sobre o desenho do estudo, abordagem computacional utilizada, medidas de resultado, principais descobertas (validadas ou não validadas), características da interface e limitações. Como produto final, um resumo comparativo (consulte a Tabela 3-1) apresenta as limitações dessas propostas.

3.2 Abordagens de avaliação de DER

Pelo menos duas abordagens processadas por computador são utilizadas na avaliação automática de DER em ambientes virtuais: sistema tutores inteligentes (STI) e processamento de linguagem natural (PLN). Ambas as técnicas começaram a ser utilizadas no processo de ensino na década de 1960 e são, resumidamente, descritas a seguir.

3.2.1 Sistema tutores inteligentes

Os STI podem ser usados para simular a tutoria pessoal um-para-um (Zawacki-Richter, Marín, Bond, & Gouverneur, 2019). Os STI visam fornecer uma instrução ou *feedback* imediato, personalizado, semelhante ao tutor humano na educação tradicional (Ognjenovic, 2017). O seu objetivo comum é possibilitar a aprendizagem de maneira significativa e eficaz, recorrendo a uma variedade de tecnologias informáticas. As funcionalidades dos STI geralmente incluem: o modelo do conhecimento do domínio, o modelo do material didático, o acesso ao material didático, o processo de aprendizagem e ensino, o teste e a avaliação do conhecimento e a administração do sistema. O processo de desenvolvimento de STI geralmente inclui modelação colaborativa do conhecimento do domínio, começando pela descrição do conhecimento em linguagem natural do especialista numa forma de conceitos e relações, formando ao mesmo tempo o inventário da ontologia de domínio (Ognjenovic, 2017). Existem diversos estudos que abordam várias aplicações de STI, eles podem ser classificados como:

ensino do conteúdo do curso, diagnóstico de pontos fortes ou fracos no conhecimento dos alunos e fornecimento de *feedback* automático, zelo pelos materiais de aprendizagem com base nas necessidades dos alunos e facilitar a colaboração entre os alunos (Zawacki-Richter et al., 2019)

3.2.2 Processamento de linguagem natural

O PLN é uma técnica de IA para extrair os elementos de interesse das informações em texto sem formatação (Nazir, Butt, Anwar, & Khan Khattak, 2017). O PLN também é visto como campo de estudo complementar no qual os computadores podem ser utilizados para analisar grandes quantidades de dados úteis sobre a linguagem humana (falada ou escrita). As aplicações de PLN são abrangentes em várias disciplinas, desde a capacidade de conversar com dispositivos eletrônicos (Ashraf, Iqbal, Khattak, & Zaidi, 2010) até a investigação básica em inglês e retórica (Hogenboom, Frasinca, De Jong, & Kaymak, 2015). PLN pode ser também utilizado para processar os requisitos iniciais de software, a fim de atingir / extrair os modelos conceptuais (Lucassen et al., 2017). Para as investigações de PLN que lidam com a interpretação da linguagem, um dos objetivos fundamentais é o de uma máquina ser capaz de entender e interpretar corretamente nuances, múltiplos significados e inferências nos padrões da linguagem humana que a mente humana pode interpretar facilmente (Berdanier, Baker, Wang, & McComb, 2019). Portanto, o desenvolvimento de conjuntos de formações apropriadas com dados qualitativos, torna-se difícil.

3.3 Avaliação dos ambientes

Foram encontradas poucas investigações de ferramentas educacionais que se propunham avaliar automaticamente o DER e algumas delas estão desfasadas ou descontinuadas. Muitas investigações sobre DER na educação são de ferramentas CASE, por exemplo (L. Yang & Cao, 2016), ou de desenhos de DER mas sem avaliação automática, por exemplo COLER (González & Suthers, 2000), CODASYS (Antony & Batra, 2002), MIST (Dimitrieski et al., 2014) e #EER (Bogdanović et al., 2008). Entretanto, estas ferramentas são limitadas pelo facto de não oferecerem a avaliação automática e, por isso, estão fora do propósito desta investigação. As próximas secções apresentam a revisão da literatura de ambientes de ensino e aprendizagem com avaliação automática de DER, destacando os métodos utilizados na avaliação e apresentando os pontos fortes e fracos de cada uma das soluções.

3.3.1 OpenMark

Thomas (2013) desenvolveu a ferramenta OpenMark, que é uma versão mais moderna do ERD-Drawing (Thomas, Smith, & Waugh, 2008) que proporciona uma avaliação automática de DER comparada à avaliação humana, e evidencia a eficácia na aplicação em grande número de DER. Num teste composto por 394 diagramas, apenas três foram avaliados incorretamente dentro da escala de 6 pontos (escalas 0.0, 0.5, 1.0, 1.5, 2.0, e 2.5) (Thomas, 2013).

Nesta nova versão, a Universidade Aberta na Inglaterra expandiu o mecanismo de avaliação de DER para o ensino virtual, que pode ser utilizado na modalidade de treino ou avaliação sumativa. O mecanismo avalia automaticamente o diagrama de um aluno e fornece *feedback* instantâneo. A nova versão é integrada no Moodle, que gere as turmas e as questões, e disponibiliza uma interface de acesso à ferramenta para os utilizadores submeterem uma resposta de DER (ver Figura 3-1).

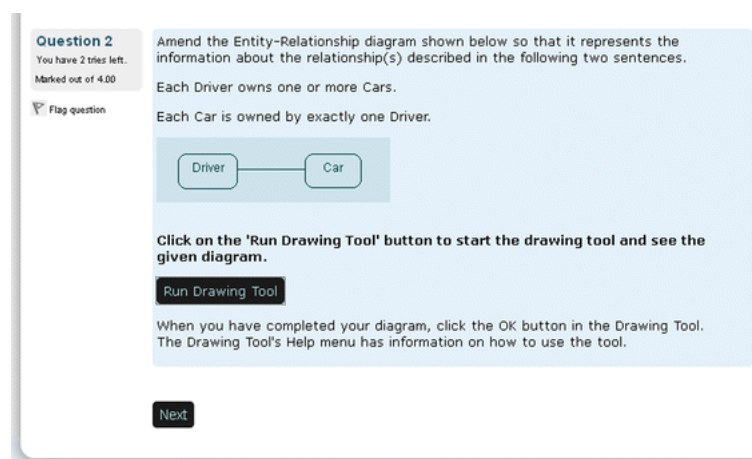


Figura 3-1. Visão do aluno para ler uma questão no Moodle e aceder a ferramenta OpenMark (Thomas, 2013).

A Figura 3-1 ilustra um exemplo da visão do aluno para submeter um DER utilizando a ferramenta. Essa interface é um *applet* que possui três itens: uma barra de menu com um conjunto de ícones de atalhos, a descrição da questão, e uma área de edição de desenho. Nesta interface o aluno pode criar uma resposta DER utilizando o menu da barra de ferramentas e submetê-la usando o botão OK (ver Figura 3-2).

Anteriormente, Thomas (2008) apresentou o método de avaliação automática do DER, que é baseado em heurística e fundamentos de PLN (*stemming* e *edit distance*). O método é dividido em cinco etapas denominadas segmentação, assimilação, identificação, agregação e

integração. As duas primeiras etapas traduzem uma imagem baseada em varredura num conjunto de primitivas diagramáticas, como caixas, linhas e texto (Thomas et al., 2008).

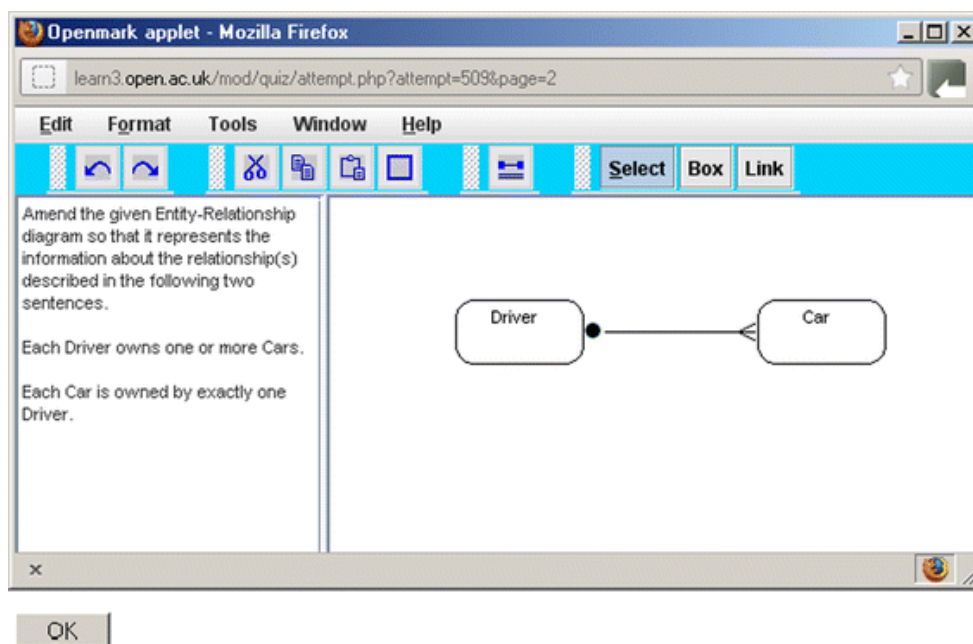


Figura 3-2. A ferramenta OpenMark na visão do aluno (Thomas, 2013)

A identificação, procura mapear o desenho do DER em forma de caixas e linhas em relacionamentos de entidades, que são denominados de unidades significativas mínimas (MMUs). Na agregação, os MMUs são combinados para contabilizar a equivalência entre o DER do aluno e a resposta do professor. Ao menos, a equivalência da relação *muitos para muitos* por um par equivalente, é detetável pelo avaliador automático (Thomas et al., 2008).

Conforme Thomas et al. (2008), a interpretação compara os DER da resposta do aluno com a resposta do professor usando uma medida de similaridade, um número real entre 0 e 1, onde o número 1 representa a resposta desejada. A correspondência dos textos usa ideias de PLN para calcular uma medida de similaridade (*stemming* e *edit distance*), que trata nomes de entidade como substantivos e relacionamentos como verbo, e as comparações de palavras após remover terminações de gerúndio e participípio.

A vantagem deste tipo de abordagem é que as palavras em frases são descartadas, bem como o uso de palavras erradas. A desvantagem ocorre quando o aluno usa palavras diferentes para descrever uma entidade ou relacionamento. O uso de sinónimos tenta resolver o problema da resposta do professor não possuir as mesmas palavras que podem ser usadas pelos alunos. Isso serve para preencher as possíveis deficiências da medida de similaridade, se for insuficiente para comparar a resposta do aluno e a resposta do sistema. Esta abordagem é muito interessante, pois pode resolver o problema da utilização de palavras inapropriadas para

identificar uma entidade. No entanto, essa abordagem pode distorcer completamente a resposta do aluno, permitindo que a sua resposta, que é realmente errada, seja considerada correta (Thomas et al., 2008).

Em resumo, a investigação apresenta uma abordagem para avaliação automática de DER, com fundamentos de PLN (Manning & Schütze, 1999) para identificar sinónimos e erros ortográficos (Thomas et al., 2005). Não somente são utilizadas a associação e as relações de DER para identificar subestruturas equivalentes em diagramas, mas também estruturas maiores, denominadas padrões (Thomas et al., 2006). Este conceito de padrão é introduzido para lidar com soluções alternativas, equivalentes da mesma questão. Os resultados são próximos aos de um avaliador humano, com uma *precisão* de 91% (Thomas et al., 2008). A ferramenta foi desenvolvida em JAVA para ser usada em desktop e incorporada ao Moodle, e depois a usabilidade foi testada por 28 alunos, cuja maioria recomenda o uso do sistema (Thomas, 2013). Este autor alega que existem alguns diagramas que não são manipulados suficientemente bem pelo algoritmo de avaliação automática, mas não o especifica.

A notação oferecida pelo OpenMark é uma variação desconhecida do *crow's foot* que está limitada apenas na descrição do nome das entidades e relacionamentos. Essa notação não possui todas características de um DER, como atributos, domínio, e identificação adequada da cardinalidade da relação. A limitação da notação oferecida pelo OpenMark simplifica demasiado o processo de avaliação, uma vez que não considera os atributos e domínios e as complexidades para avaliar outros conceitos derivados dos componentes do DER, como atributos multi-valor, composição, atributos derivados, domínio, chave, entidades fracas, fortes, entre outros. Portanto, por mais sofisticado que seja o algoritmo de avaliação, ele está reduzido à comparação de nomes das entidades e seus relacionamentos.

3.3.2 EER-Tutor

O EER-Tutor é um sistema tutor inteligente (STI) que usa a modelação baseada em restrições para modelar o conhecimento do domínio e gerar modelos de alunos. Foi desenvolvido para que estudantes universitários aprendessem o projeto de BD conceptual usando entidade e relacionamento estendida (Elmadani et al., 2015). Esta ferramenta tem sido desenvolvida por Antonija Mitrovic ao longo de quase 20 anos. A ferramenta original pode ser encontrada com o nome KERMIT (Suraweera & Mitrovic, 2002) e também como ER-Tutor (Suraweera & Mitrovic, 2004). Diversas investigações foram feitas por Mitrovic dedicadas não somente à automação do DER, mas também a SQL, UML e linguagem de programação, pelo

que possui outras ferramentas baseado em STI denominadas, SQL-Tutor, COLLECT-UML, J-LATTE, NORMIT, ERM-Tutor (X. Chen, Mitrovic, & Mathews, 2018; Mitrovic, 2012).

Atualmente, O EER-Tutor usa uma arquitetura web (ver Figura 3-3), como resultado dos esforços para reimplementar o ER-Tutor e o KERMIT. O EER-Tutor explora o efeito do *feedback* instrucional projetado de acordo com os princípios da arquitetura cognitiva e da teoria da aprendizagem. Como um sistema de software, o EER-Tutor é baseado numa variedade de tecnologias e linguagens de programação: Lisp, Java, HTML, XML e o protocolo HTTP na camada de aplicação (Elmadani et al., 2015). O livro de Elmasri e Navathe (2015) descreve pormenorizadamente a notação do DER utilizada no EER-Tutor.

The screenshot displays the EER-Tutor web interface within a Microsoft Internet Explorer browser window. The interface is divided into several sections:

- Problem Statement:** Located at the top left, it contains a text-based problem description for a database design task involving staff information.
- Navigation Frame:** A horizontal bar below the problem statement containing buttons for "Next Problem", "History", "Student Model", "Tutorial", "Help", "Print", and "Log Out".
- Drawing Area:** The central workspace where an Entity-Relationship (ER) diagram is being constructed. The diagram features an entity "EMPLOYEE" with attributes "Date_of_birth", "Name", "Id", "Gender", and "Address". It is connected via a disjoint specialization relationship "d" to four sub-entities: "MANAGER", "ENGINEER", "TECHNICIAN", and "SECRETARY". "MANAGER" has a relationship "CONTROLS" with "PROJECT" (attributes: "Project number", "Project name"). "ENGINEER" has attributes "Type" and "Grade". "TECHNICIAN" has attribute "Grade". "SECRETARY" has attribute "Typing_speed". "EMPLOYEE" is also connected via another disjoint specialization relationship "d" to "HOURLY" (attributes: "Pay_scale") and "SALARIED" (attribute: "Salary").
- Feedback Frame:** Located on the right side, it provides instructional text explaining the function of each navigation button and offers guidance on how to draw the ER diagram.
- Submission Frame:** At the bottom, it contains buttons for "Submit Answer" and "Show Full Solution".

Figura 3-3. Interface ERR-Tutor.

Na versão web do EER-Tutor, foi explorada uma nova forma de *feedback*, projetada com os princípios da arquitetura cognitiva, de forma mais eficaz do que as mensagens de *feedback* convencionais. Nesta versão, existem 57 questões registradas, que são apresentadas na parte superior da interface, seguido de uma barra de ferramentas com os componentes apropriados da notação do DER. Os alunos, ao submeterem as soluções, recebem o *feedback* mostrado no

painel direito da interface. O sistema também destaca em vermelho partes incorretas da solução na área de desenho. Existem destaques visuais nas descrições das questões, que são palavras coloridas na questão. Os destaques visuais basicamente listam quais são as entidades, atributos e relacionamentos do problema a ser resolvido pelo aluno (Mitrovic & Suraweera, 2016).

Conforme Mitrovic e Suraweera (2016), dois componentes de aprendizagem, detecção de erros e correção de erros são subtilmente abordados nas mensagens de *feedback* no EER-Tutor. O sistema compara as soluções do aluno à solução ideal usando o conhecimento de domínio representado na forma de restrições (mais de 200 regras), que são classificadas em sintáticas e semânticas. Restrições semânticas permitem que o sistema lide com soluções alternativas para alunos, ou seja, as equivalências do DER do aluno e da resposta do professor são identificadas por regras.

Ao contrário da proposta da ferramenta que pretendeu dar um *feedback* baseado na intuição usando STI, na nova versão o sistema simula uma avaliação do DER que dá uma combinação de *feedback* para detecção e correção do erro. Esse *feedback* baseia-se na teoria de aprender com os erros de desempenho Ohlsson (1994), que afirma que “o papel do *feedback* deve ser o de identificar o erro e explicar o que constitui o erro (atribuição de culpa), bem como repetir o princípio do domínio violado pela solução do aluno (correção)” (Ohlsson, 1994). Porém, no EER-Tutor todas as regras de avaliação do DER incluindo os *feedbacks* devem ser estaticamente definidas no código-fonte (Mitrovic & Suraweera, 2016).

Em resumo, o EER-tutor usa um STI e modelação baseada em restrições para avaliar automaticamente o DER, cerca de 200 regras avaliam a sintaxe e a semântica dos diagramas do aluno contra a resposta do professor. A notação do DER é uma versão estendida do Chen. As desvantagens de qualquer STI estão nas regras, seja na quantidade de regras a serem implementadas, bem como nas tomadas de decisões que podem ser diferentes de professor para professor. É muito difícil prever todos os erros que os alunos podem cometer ao submeter as suas respostas e escrever todas as formas de *feedback* para cada um desses casos, não é uma solução viável. Além disso, o destaque visual das questões torna a resposta restritiva, uma vez que o aluno reconhece visualmente todos os itens que compõem a resposta esperada pelo sistema. Mais, numa mudança de notação, as regras não poderiam ser reaproveitadas.

3.3.3 ERM-VLE

O ERM-VLE é um sistema cliente/servidor, que visa disponibilizar a aprendizagem do modelo ER e fornecer um *feedback* imediato aos alunos sobre a qualidade de seus modelos. A

abordagem visa ajudar os alunos a distinguir entidades de atributos e relacionamentos a partir da descrição do problema, onde o aluno deve modelar uma BD para um determinado problema, navegando no mundo virtual e manipulando objetos. O mundo virtual contém diferentes tipos de salas, como salas de criação de entidades e salas de criação de relacionamentos. Os autores afirmam que a organização do ambiente reflete a estrutura da tarefa e incentiva um padrão de navegação no mundo virtual. O ambiente obriga os alunos a inserirem um projeto de BD de forma metodológica. Para criar um componente, o aluno deve usar uma frase do cenário que justifique a existência desse componente (ver Figura 3-4). Os alunos podem nomear os seus componentes numa caixa de texto, mas não podem desenhar usando o rato com funções de clicar e arrastar. Por exemplo, um aluno cria uma entidade enquanto está na sala de criação da entidade. O comando de avaliação fornece dicas para modelar o esquema de ER (Hall & Gordon, 1998).

A interface do ERM-VLE (ver Figura 3-4) consiste em três painéis. O painel “Cenário” contém os requisitos para a base de dados que está sendo modelada. O painel “ERM atual” fornece uma representação gráfica do modelo de ER que o utilizador está a construir. A representação gráfica é atualizada dinamicamente para refletir as atividades do aluno, mas o aluno não interage diretamente com a representação gráfica. O aluno interage com o mundo virtual apenas emitindo comandos de texto. O painel “mundo do ERM” contém um registo de interações passadas entre o aluno e o mundo. A solução de cada problema está incorporada no mundo virtual. As correspondências entre as frases do cenário e as construções do modelo ER são armazenadas na solução. O aluno só pode estabelecer as correspondências ideais do sistema. Se o aluno tentar estabelecer uma associação que não esteja em conformidade com a solução do sistema, este intervém e informa o aluno que a associação não é permitida. O sistema foi avaliado por um grupo de especialistas e alunos iniciantes em BD, os especialistas notaram que a estrutura do mundo virtual os tinha restringido (Hall & Gordon, 1998). Por outro lado, os alunos notaram que tinham aumentado a compreensão do modelo ER. No entanto, esses comentários não podem ser tratados como evidências substanciais quanto à eficácia do sistema, uma vez que o sistema não foi avaliado adequadamente (Suraweera & Mitrovic, 2004).

O método de avaliação do sistema tem apenas uma solução correta para cada cenário e os alunos só podem produzir essa solução. Qualquer tentativa errada por parte dos alunos é aprisionada e o *feedback* imediato é dado com base na solução do sistema. Isso torna o sistema inflexível e inviável para soluções alternativas. Este método tem uma alta tendência para incentivar a aprendizagem superficial, pois os utilizadores são impedidos de cometer erros e

não recebem explicações sobre os seus erros. O sistema foi avaliado com um grupo de projetistas experientes e novatos, projetistas experientes sentiram que a estrutura do mundo virtual os restringia (Hall & Gordon, 1998). Além disso, um ambiente de ensino e aprendizagem baseado em texto é um ambiente altamente artificial no qual se constroem DER. Alunos que aprendem o DER utilizando o ERM-VLE teriam dificuldade em se habituar a projetar a BD fora do ambiente virtual (Suraweera & Mitrovic, 2004).

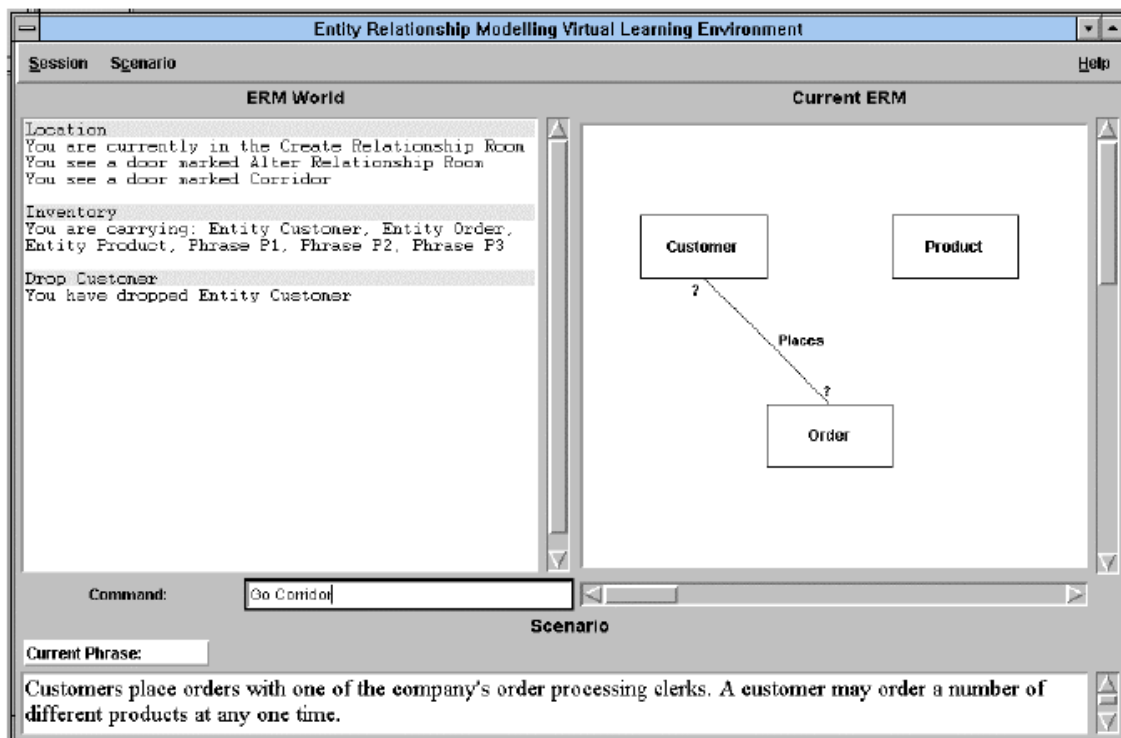


Figura 3-4. Interface VLE-ERM.

Em resumo, o ERM-VLE é um sistema baseado em texto para projetar o modelo ER, no qual os alunos projetam BD navegando no mundo virtual. O mundo virtual consiste em salas, como salas de criação de entidades e salas de criação de relacionamentos, e os alunos escrevem comandos de texto, como escolher, soltar, nomear, avaliar, criar e remover para manipular os objetos. Essa ferramenta provou ser muito restritiva e os alunos relataram desânimo por serem forçados a seguir o caminho da solução ideal (Hall & Gordon, 1998). Os pontos negativos são: a notação do DER utilizada é desconhecida e não oferece o uso de atributos bem como a identificação dos diversos conceitos de DER. A resposta esperada pelo aluno deve ser igual à registrada no sistema, ou seja, é restritiva e não há qualquer solução para o problema de DER equivalentes, bem como não há menção de uso de algoritmos para pontuar ou classificar a resposta do aluno. O *feedback* não fornece informações de erros da resposta submetida pelo aluno e não existe associação com os conceitos de DER. Não é utilizado nas avaliações formativas ou sumativas.

3.3.4 DATSys

O *framework* DATSys executa avaliação em diagramas e pode ser usado em diferentes contextos, por exemplo, em DER ou UML (Tsintsifas, 2002). DATSys foi adaptado por Higgins e Bligh (2006) e usado para avaliação do DER e alguns problemas foram relatados (C. A. Higgins & Bligh, 2006). O método utilizado no DATSys avalia as submissões dos alunos, comparando as suas soluções contra um conjunto de ferramentas usando métricas estáticas e dinâmicas. As métricas estáticas são responsáveis por examinar, por exemplo, tipografia, complexidade e estrutura do programa. As métricas dinâmicas usam regras predefinidas pelo professor para estimar a nota final. Os dois tipos de métricas de avaliação usam um reconhecedor de expressão para verificar a presença (ou ausência) de *tokens* definidos (C. A. Higgins & Bligh, 2006).

Então, quando o subsistema de avaliação executa a ferramenta fluxograma, o fluxograma do aluno é convertido em código BASIC, que é então alimentado para a ferramenta dinâmica (C. A. Higgins & Bligh, 2006). A ferramenta dinâmica entende que o DER é composto por nós, onde cada nó é identificado por dois atributos, nome (que pode ser entidade, relação e atributo) e o conteúdo do texto. Uma expressão regular compara os nós existentes no DER da resposta do sistema contra os nós do DER da resposta do aluno. A abordagem da avaliação é baseada em nós do diagrama, na qual os tipos de nós e suas conexões são avaliados de acordo com os critérios fornecidos pelo professor. Posteriormente, a ferramenta dinâmica executa o programa BASIC e devolve a nota e o *feedback* aos alunos. No entanto, vários problemas foram relatados, o primeiro mostrou que esse método ajuda demasiadamente os alunos a obterem as soluções corretas. O segundo problema identificado é que este método não aborda as várias soluções de modelo igualmente válidas com características ligeiramente diferentes e mutuamente exclusivas, ou seja, não consegue avaliar a equivalência entre DER. O terceiro, é que alguns *feedbacks* eram confusos, fornecendo um *feedback* positivo enquanto a solução do DER do aluno continha alguns erros, neste caso nem o assistente do laboratório conseguia identificar o erro. Um quarto problema foi identificado, os *feedbacks* não instruíam os alunos para a identificação e correção dos erros (C. Higgins, Symeonidis, & Tsintsifas, 2002).

Em resumo, DATSys fornece um *framework* que visa automatizar o processo de avaliação para diversos tipos de diagramas, e conseqüentemente, possui um mecanismo genérico baseado em nós (identificação do nome das entidades, atributos e relacionamentos) e expressões regulares para fornecer a avaliação automática. A partir do DER da resposta do professor uma mensagem do tipo certo ou errado é enviado para cada nó do DER da resposta

do aluno. Ou seja, essa abordagem não avalia a equivalência entre os DER do sistema e aluno. A pontuação final é estimada a partir de critérios previamente definidos pelo professor, que verifica a presença ou ausência dos nós do DER do aluno em comparação com o DER do professor. A notação utilizada do DER é uma variação do Chen, como pode ser notado nos relacionamentos e atributos do DER da Figura 3-5. As diversas falhas relatadas pelos alunos têm origem na abordagem de avaliação automática do DER que é restrita, por outras palavras, força o aluno a seguir a resposta do professor. Portanto, a abordagem não consegue dar um *feedback* para identificação e correção de erro.

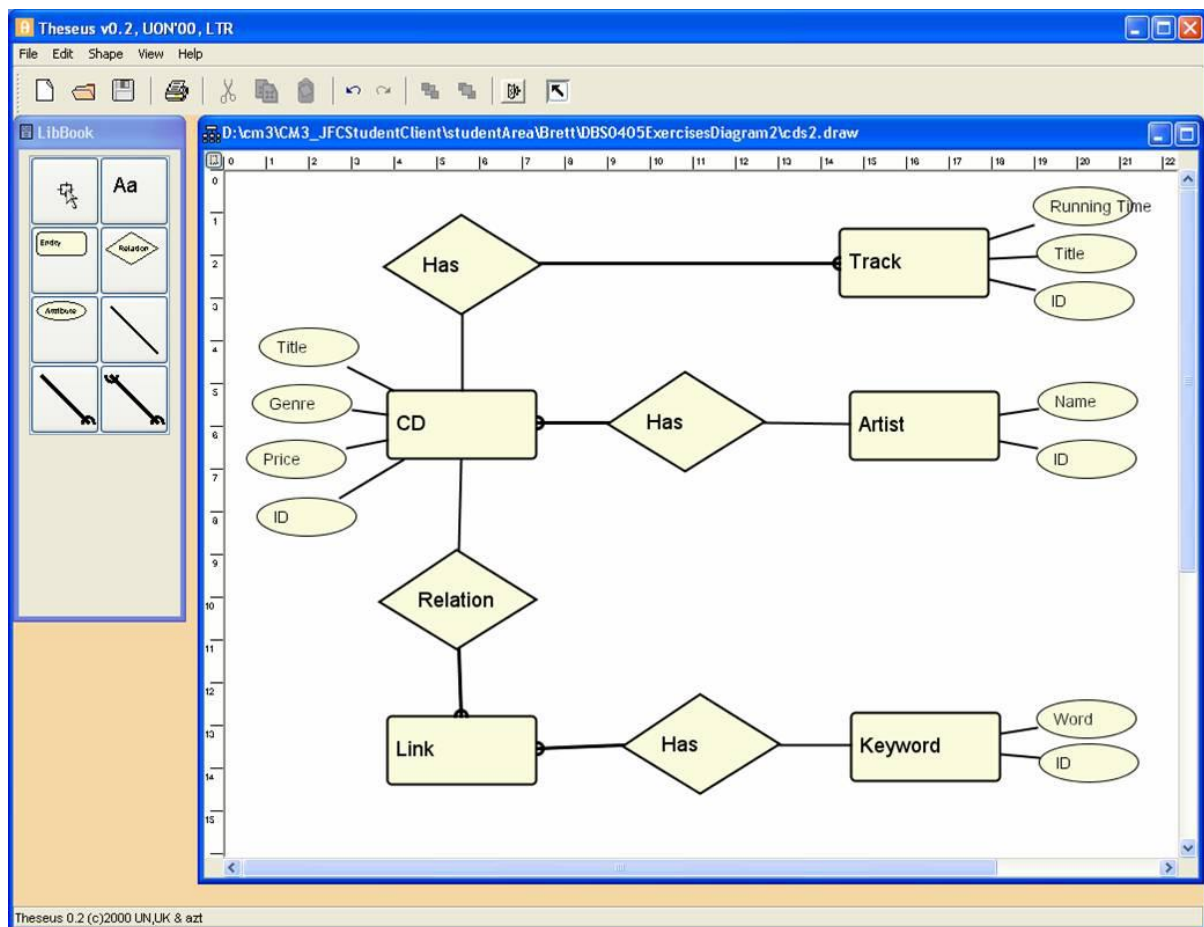


Figura 3-5. Interface do DATsys com a resposta DER do aluno.

3.3.5 Principais problemas

As ferramentas de avaliação automática de DER, ao contrário das ferramentas CASE tradicionais, são desenvolvidas especificamente para iniciantes na modelação de ER. Mesmo que o objetivo inicial fosse construir sistemas que permitissem aos iniciantes construir a BD simplesmente especificando as suas necessidades, o objetivo da avaliação automática do DER ainda deve ser plenamente realizado. Além disso, as ferramentas CASE ainda não podem, e talvez nunca consigam, substituir completamente um desenvolvedor humano de BD, porque

as ferramentas não conseguem entender e raciocinar completamente sobre o domínio do aplicativo. Essas limitações levaram à exploração de métodos e ferramentas mais eficazes para o ensino de modelação de BD.

Ambientes com STI para ensinar modelação de BD podem ser usados como ferramentas auxiliares de ensino para iniciantes em modelação de BD. Uma grande vantagem desses sistemas é que permitem que os alunos aprendam, ao seu próprio ritmo, de modo diferente do ensino presencial em sala de aula. As ferramentas de software não inteligentes, embora sejam mais fáceis de desenvolver, têm eficácia limitada, pois foram projetadas para um aluno ou um professor e não têm a capacidade de aprender e usar modelos para avaliar automaticamente um DER e oferecer um *feedback* personalizado.

Os STI são projetados para automatizar o ensino individual e podem ser considerados uma forma mais eficaz de ensinar. O STI oferece maior flexibilidade em contraste com os tutores de software não inteligentes, já que eles se podem adaptar a cada aluno, individualmente. Estudos empíricos realizados para avaliar o STI noutros domínios mostraram grandes melhorias na aprendizagem dos alunos. Embora o STI se tenha mostrado eficaz em vários domínios, um STI efetivo para modelação de BD ainda não foi desenvolvido.

O ERM-VLE, ambiente baseado em texto para modelo ER, é um ambiente altamente artificial para a construção de DER. Um aluno teria dificuldade em transferir os conhecimentos adquiridos utilizando o ERM-VLE para modelar uma BD para requisitos da vida real. Como as soluções são incorporadas no próprio ambiente virtual, os alunos que usaram o sistema reclamaram que era muito restritivo, pois foram forçados a seguir o caminho da solução ideal. O método de forçar o utilizador a seguir o caminho da solução do sistema, aumenta o risco de incentivar a aprendizagem superficial. A abordagem concentra-se em avaliar os diagramas sem obter ajuda dos alunos. Os alunos têm suporte do ambiente Moodle para visualizar e responder os problemas de DER. Por outro lado, a abordagem do ERM-VLE justifica o envolvimento do aluno no processo educacional, mas força-os a seguirem o modelo de projeto de BD à medida que inserem a sua solução no sistema.

Apenas dois projetos de investigação abordam o problema de soluções equivalentes. O ERM-VLE propõe um conceito chamado “padrão” para lidar com soluções alternativas. Por exemplo, um padrão específico é usado para lidar com diferentes representações de relacionamentos “muitos para muitos”. O EER-Tutor propõe restrições semânticas e produziu um conjunto de restrições para avaliar automaticamente os DER.

As ferramentas EER-Tutor e OpenMark são exemplos de projetos em funcionamento, com longo período de utilização. Até agora, eles desenvolveram algumas aplicações práticas com algumas restrições. Ambas conseguiram a automação da avaliação dos DER, porém simplificam as questões ou restringem as soluções dos alunos. Por exemplo, questões com uma solução válida são solicitadas e uma lista limitada de nomes de componentes, para soluções dos alunos, é aceita. Uma nova abordagem, cobrindo mais tipos de problemas e aplicando menos restrições às soluções dos alunos será, educacionalmente, mais aceitável.

A automação completa requer a incorporação de um conjunto completo de regras e de conhecimento necessário sobre os problemas no sistema. Embora isso seja possível para uma questão individual, requer aumentar a carga de trabalho do professor de forma inaceitável. Por exemplo, o professor pode ser solicitado para inserir todas as soluções possíveis para uma questão. No entanto, isso não é prático para o professor.

As soluções computacionais adotadas para avaliar automaticamente DER nesses ambientes apresentam as seguintes limitações:

- **Interface restritiva** ocorre quando o sistema não permite ao utilizador criar o DER utilizando as funções do rato de clicar e arrastar;
- **Resposta restritiva** ocorre quando os alunos têm limitações em responder aos diagramas porque o ambiente aceita apenas uma solução correta, consequentemente os alunos são direcionados a seguir uma resposta padrão através da interface gráfica, ocasionando um *feedback* do tipo certo ou errado;
- **Destaques visuais** ocorrem quando os sistemas usam realces visuais que simplificam demasiadamente o problema destacando as palavras que fazem parte da resposta esperada do sistema, limitando assim a reflexão do aluno para uma resposta;
- **Notação restritiva** ocorre quando o sistema oferece uma notação de DER desconhecida ou uma notação sem todas as propriedades do DER, o que implica um conjunto reduzido de conceitos de DER que são avaliados e *feedbacks*;
- **Avaliação sem Equivalência** ocorre quando o sistema não faz o tratamento da equivalência entre os DER;
- **Feedback inadequado** ocorre quando os alunos recebem um *feedback* positivo enquanto há erros na solução do DER;

- **Feedback estático** ocorre no sistema especialista que usa uma abordagem limitada por um conjunto de regras pré-estabelecidas, portanto, o *feedback* é estático para casos de detecção e correção de erros.

A Tabela 3-1 lista as quatro abordagens de avaliação automática de DER. A primeira coluna identifica o nome do ambiente avaliado e o autor; a segunda coluna, chamada flexibilidade do projeto, define se o ambiente fornece aos utilizadores a opção de desenhar o DER usando as funções de clicar e arrastar do rato. Em seguida a coluna normalização, define se a solução suporta normalização. Em seguida, a coluna de ensino online indica se o sistema pode ser usado via internet. A seguir, a coluna de progresso indica se a solução monitoriza o progresso dos alunos no desenvolvimento de suas habilidades de modelo ER. A seguir, a coluna de tecnologia, identifica a linguagem de programação em que foi desenvolvida. Em seguida, as colunas vantagens e desvantagens descrevem os pontos fortes e fracos de cada solução e, em seguida, a coluna notação, descreve em que notação a abordagem foi aplicada. A última coluna indica se a proposta foi validada por uma experiência controlada.

Tabela 3-1. Características das ferramentas de avaliação de DER.

<i>Ambientes</i>	<i>Flexível</i>	<i>Forma Normal</i>	<i>Online</i>	<i>Progresso</i>	<i>Avaliação Automática</i>	<i>Tecnologia</i>	<i>Pontos Fortes</i>	<i>Pontos Fracos</i>	<i>Notação</i>	<i>Validado</i>
<i>ERR-Tutor (Mitrovic & Suraweera, 2016)</i>	S	N	S	S	Tutor	Java	<i>Feedback</i> automático, Avaliação formativa	Destaques visuais, <i>Feedback</i> estático	EER	S
<i>ERM-VLE (Hall & Gordon, 1998)</i>	N	N	N	S	Heurística	C	Pode ser usado colaborativamente nas atividades de ensino	Interface restritiva, Resposta restritiva, Avaliação sem Equivalência	Desconhecida	S
<i>DATsys (Bligh, 2007)</i>	N	N	N	S	Expressões regulares	Java	<i>Feedback</i> automático, pode ser usado na avaliação formativa e sumativa	Resposta restritiva, <i>Feedback</i> inadequado, Avaliação sem Equivalência	Variação do Chen	S
<i>OpenMark (Thomas, 2013)</i>	S	N	S	S	PLN + Heurística	Java	Avaliação formativa e sumativa no Moodle	Notação restritiva	Desconhecida	S

Framework de Avaliação Automática de DER e SQL

4.1 Considerações iniciais

Este capítulo apresenta o *framework* de avaliação automática de DER e SQL é baseado em compiladores, métricas de engenharia de software e aprendizagem computacional. O *framework* tem como entrada a resposta do aluno e a resposta do professor, que pode ser em DER ou SQL, e devolve como saída, uma nota ou a classificação final acompanhada de um *feedback* semântico. Este *framework* foi implementado no ambiente LabDER para retornar o *feedback* aos usuários sempre que submeterem ou visualizarem uma resposta.

O processo de avaliação automática proposto neste *framework* é constituído por três partes com objetivos bem definidos. O primeiro nível oferece mecanismos para identificar, comparar e mapear cada elemento de um diagrama ER em conceitos de BD. O segundo nível, tem o objetivo de fornecer indicadores para entender e avaliar cada elemento e conceito de BD identificados no diagrama ER. O objetivo do terceiro nível é encontrar um modelo preditivo capaz de avaliar um diagrama ER e que mais se aproxime de um avaliador humano.

Essa abordagem compara a distância entre as respostas do aluno e as do professor, e fornece um *feedback* semântico misturando os resultados da análise do compilador com as métricas, enquanto prevê a classificação final da resposta do aluno através da aprendizagem computacional.

4.2 Breve histórico

A primeira versão do *framework* foi desenvolvida em 2007 para o LabSQL, como resultado de investigação de mestrado, que teve como objetivo desenvolver uma solução computacional para avaliar automaticamente questões de SQL (A. D. P. Lino, 2007). Nesta versão, a abordagem de avaliação automática do LabSQL prediz a avaliação do professor em

escala de percentagem por meio de regressão linear múltipla e métricas de engenharia de software. O processo de análise de regressão teve como entrada os dados de avaliações de especialistas, sendo o modelo encontrado após sete meses de exaustivos testes e observações utilizando o software Microsoft Excel.

O *framework* da versão de 2013 implementada no LabSQL analisava a sintaxe da resposta SQL do aluno em diferentes níveis, emitindo um *feedback* com o resultado da consulta executada pelo interpretador do SGBD e dicas para correção ou melhoria da resposta do aluno baseado em métricas de engenharia de software (ver Figura 4-1). Nesta versão, o *framework* resolve o propósito da avaliação automática de SQL, mas a solução não é genérica o suficiente para ser adaptada para avaliar questões de outro domínio de conhecimento, por exemplo Java.

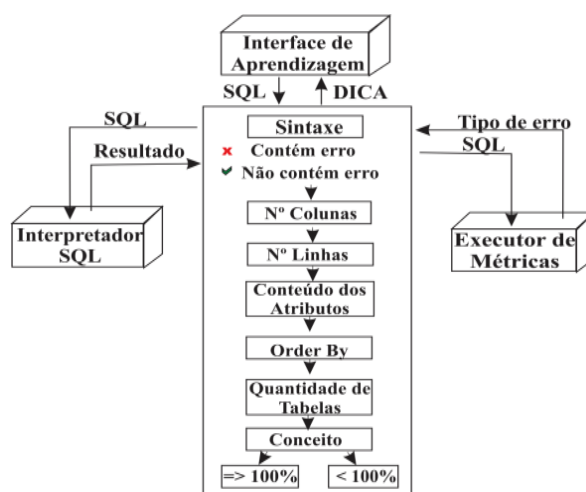


Figura 4-1. Arquitetura do LabSQL, extraído de (Melo et al., 2013).

Os principais problemas da versão de 2013 estão relacionados à restrição da linguagem SQL e ao modelo para prever a avaliação do professor. Ainda que fosse incorporado um robusto mecanismo para compreender outras linguagens de programação, o problema da predição persistiria. Seja pela dificuldade de encontrar métricas para aferir a resposta do aluno, bem como pela dificuldade em encontrar modelos preditivos que mais se aproximem das avaliações do professor.

Em 2016, investigamos mecanismos de Inteligência Artificial (IA) e, mais especificamente, de Aprendizagem Computacional para melhorar todo o processo de concepção de modelos de avaliação automática do *framework*. Neste contexto, a primeira contribuição foi atribuir a tarefa de análises de dados ao computador para encontrar o modelo preditivo na forma de uma função objetiva. O resultado foi a publicação sobre algoritmos genéticos, que basicamente desempenhou a tarefa de encontrar a função. Para isto, replicou-se os dados de SQL para que o computador chegasse autonomamente à função que explicasse a relação entre

as avaliações dos especialistas e as métricas extraídas da resposta do aluno (Adriano Lino, Rocha, & Sizo, 2016).

Outra investigação importante que deu origem a versão final do *framework* envolveu a construção de um simulador de respostas para analisar os erros cometidos pelos alunos (ver item “c” na Figura 4-2). A arquitetura do simulador foi desenvolvida especialmente para conceber uma base de dados para analisar os erros que os alunos cometeram durante a utilização do ambiente virtual. Essa arquitetura foi concebida e baseada em compiladores, justamente pela sua grande capacidade de análises de erro.

A partir da comparação entre os compiladores (ver itens “b” e “c” da Figura 4-2), observou-se que as respostas dos alunos poderiam ser tratadas como uma linguagem qualquer, sejam um DER, SQL, ou até mesmo diagramas UML. Ao aplicar a teoria de compiladores, podem-se encontrar as equivalências entre as respostas baseadas em gramáticas regulares ou geradas pelo programa, e ainda ter uma excelente forma de análise e detecção de erros, mas sem a nota final do aluno. Assim, adicionou-se a aprendizagem computacional supervisionada para gerar o melhor modelo preditivo baseado no histórico de avaliações do professor. O resultado é um *framework* genérico, capaz de avaliar diferentes estruturas, em especial diagramas de entidade e relacionamento. Maiores detalhes sobre avaliação automática de SQL podem ser encontrados na publicação sobre desenvolvimento de um simulador para analisar os erros de SQL (A. Lino, Rocha, Macedo, & Sizo, 2019).

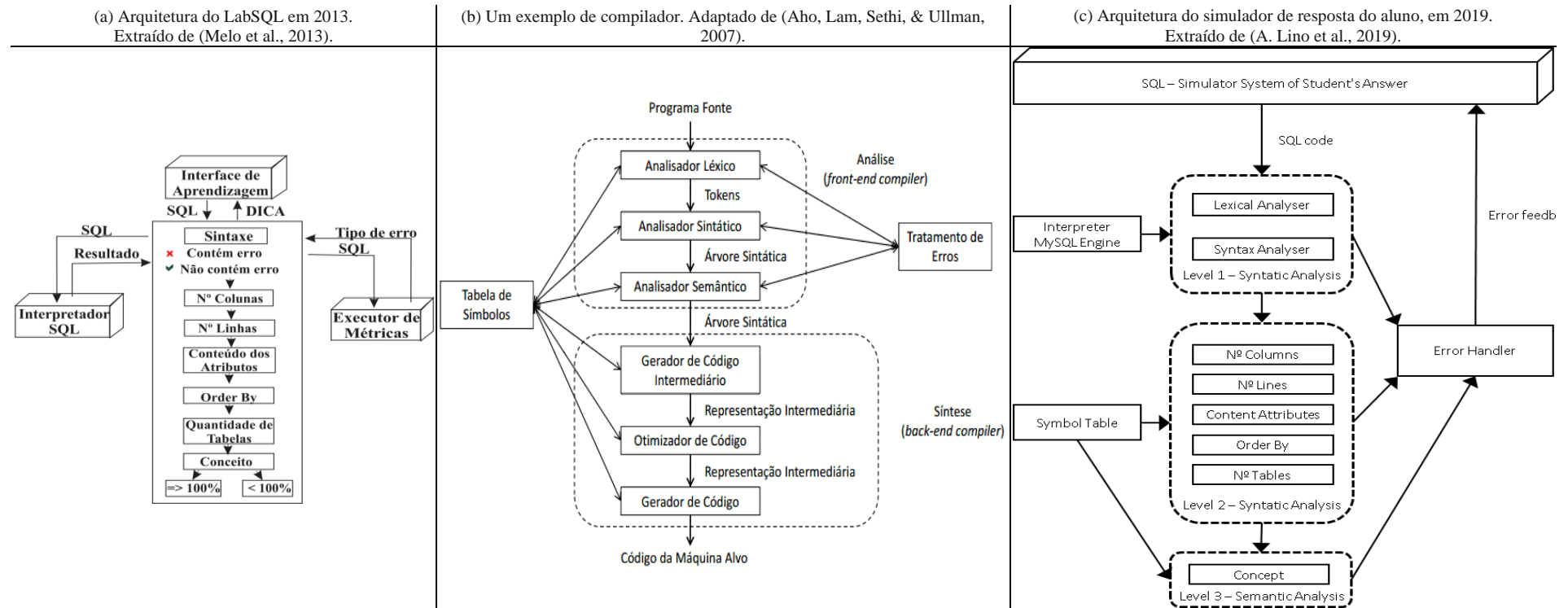


Figura 4-2. Uma comparação entre as arquiteturas do LabSQL (a), um compilador (b), e um simulador de respostas para análise de erros (c).

4.3 Abordagem da avaliação automática de DER

A abordagem de avaliação automática de DER foi estruturada em três níveis (ver Figura 4-3). No primeiro nível faz-se a análise das respostas, que é inspirada na teoria de compiladores, e contém as etapas de análise léxica, sintática e semântica, que juntas trabalham em simbiose com a tabela de símbolos e o tratamento de erro para oferecer um mecanismo eficiente de identificação dos elementos que compõe a resposta do aluno e a resposta do professor. No segundo nível, as métricas de engenharia de software são usadas para medir as estruturas de cada elemento contido nas respostas do aluno e professor, e fornecem um conjunto de indicadores para medir a distância entre as respostas, bem como o uso de conceitos DER. No terceiro nível, a predição recorre a algoritmos de aprendizagem supervisionada para aferir a resposta do aluno, e tem como parâmetros de entrada as métricas e as classificações dos DER.

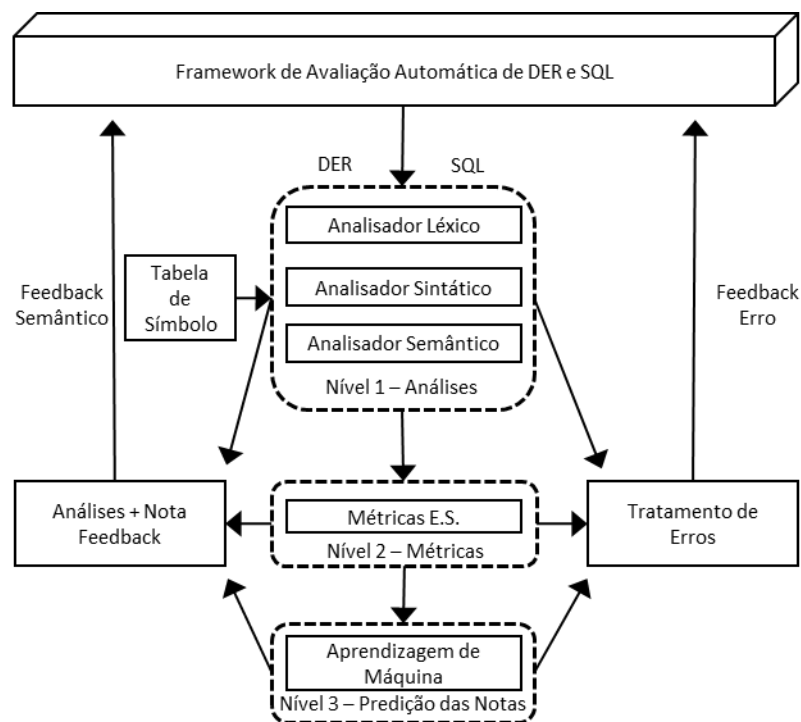


Figura 4-3. Framework de Avaliação Automática de DER e SQL.

Os compiladores são amplamente conhecidos pela sua capacidade de tradução, pois mapeiam um programa de origem num programa de destino semanticamente equivalente (Aho et al., 2007). Nesse sentido, o compilador exerce papel fundamental para o *framework*, pois através das suas diferentes análises é feita a tradução do programa escrito para uma linguagem escrita de alto nível para uma linguagem de computador.

O *framework* trata o DER como programa de alto nível e, de forma análoga a um compilador, divide o programa em partes constituintes e impõe-lhes uma estrutura gramatical. Essa estrutura é usada para detetar se, recorrendo a análises, o programa de origem (DER ou SQL) é sintaticamente malformado ou semanticamente inconsistente. Só com base nesta informação é que o sistema fornece mensagens informativas, para que o utilizador tome medidas de correção.

O nível de análise também recolhe informações sobre o programa de origem e armazena numa estrutura de dados, denominada tabela de símbolos, que é passada em conjunto com a representação intermediária para a parte de síntese. Esta constrói o programa de destino desejado da representação intermediária e as informações na tabela de símbolos.

O processo completo de um compilador pode ser visto na Figura 4-4, contendo alguma linguagem de alto nível como entrada e gera código de máquina como saída (Singh, Sharma, & Varshney, 2009).

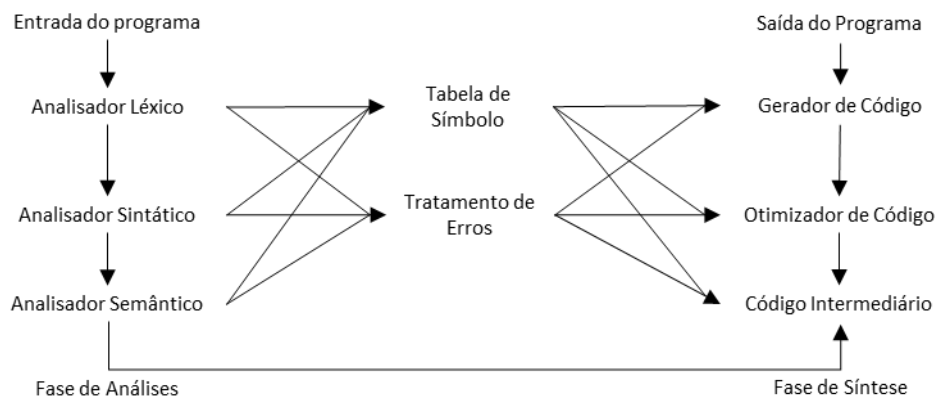


Figura 4-4. Processo de funcionamento de um compilador (Singh et al., 2009)

Este processamento inicial de entrada suporta o reconhecimento correto de um problema com o DER e capacita o sistema a fornecer o *feedback* mais apropriado. Em caso de erro na resposta do aluno, a saída inclui um *feedback* semântico com detalhes do erro que ajudam o aluno a encontrar uma solução correta. Em caso de resposta correta do aluno, há duas saídas de *feedback* semântico: (i) dicas para auxiliar o aluno a superar a resposta professor, ou (ii) uma felicitação, se a resposta do aluno for melhor do que a resposta do professor. Em ambos os casos, o *feedback* inclui a nota final gerada pelas métricas e aprendizagem computacional.

O tratamento de erros dos SGBD, normalmente associa conceitos de BD ao resultado da análise léxica, sintática e semântica, e produz um *feedback* semântico para erros comuns. Por exemplo, qualquer SGBD emitirá um *feedback* de erro ao tentar criar mais de uma tabela com o mesmo nome, bem como uma tentativa de projetar uma consulta com colunas ambíguas.

Assim, os erros detetáveis pelo SGBD também estão incorporados no *framework*, todos os erros são reencaminhados na forma de *feedback*, uma vez que o objetivo não é recriar um compilador de SGBD.

Entretanto, a resposta do aluno é sempre avaliada pelo *framework* e, para ser considerada correta, não basta apenas ser interpretável pelo computador, precisa de ter uma forma computacional equivalente à resposta do professor. Então, as regras de sintaxe e semântica são impostas para obter uma equivalência entre as respostas do aluno e a resposta do sistema; o processo de análises verifica as equivalências e oferece um *feedback* semântico que podem incluir os elementos que estão em falta ou em excesso na resposta do aluno.

Diversos conceitos de BD necessários à solução computacional de DER que sejam de interesse podem ser incluídos no *framework*. Por exemplo, o conceito de homónimos pode ser detetável durante a fase de análises e então o *framework* emite um *feedback* de erro, bem como tratar como erros outros tipos de desenhos mal projetados, que podem até ser considerados como corretos pelo interpretador do SGBD, como por exemplo, tabelas sem chave primária ou sem relacionamentos.

O uso de um compilador nesta abordagem ainda resolve o problema da ambiguidade, que é uma das limitações encontradas nos ambientes de avaliação automática de DER. A partir da análise de sintaxe e de semântica, o *framework* identifica as equivalências entre as respostas do aluno e professor. No contexto dos ambientes educacionais, a identificação da equivalência é de suma importância, pois auxilia na deteção de burla, isto é, numa tentativa de simular um resultado correto com menor esforço.

Uma outra característica explorada pelo compilador é a alta eficiência, que se caracteriza pela execução dos programas através da alocação eficiente de registos (Su & Yan, 2011). Adicionalmente, a eficiência da compilação, que está associada à tarefa de indicar os erros e de se recuperar rapidamente, caso exista algum. E por fim, a eficiência escrita, que incluiu a depuração e execução dos programas para descobrir como resolver alguns problemas mais rapidamente (Su & Yan, 2011).

No *framework*, a perspectiva de alta eficiência possui dois aspetos: (1) eficiência da execução, que procura descobrir se a resposta do aluno é mais eficiente do que a resposta do professor, recorrendo às métricas de engenharia de software, e (2) eficiência de compilação, que dá primazia, ao tratamento de erros, identificando rapidamente os erros existentes na solução do aluno, a partir da resposta do professor. O *framework* combina então esses dois elementos de eficiência e entrega como saída, o que denominamos de *feedback* semântico, pelo

qual é possível identificar o erro e permite o desenvolvimento de novas soluções computacionais, seja para o caso de corrigir um erro, ou aperfeiçoar a solução.

Aho et al. (2007) define compiladores como: “Um programa que lê um programa escrito num idioma - o idioma de origem - e o traduz para um programa equivalente noutra idioma - o idioma de destino. Como parte importante deste processo de tradução, o compilador reporta ao seu utilizador a presença de erros no programa fonte.” A estrutura de compiladores é generalizável exatamente porque pode ser aplicada na transformação de qualquer linguagem, sendo o tratamento do erro a chave da identificação, se a estrutura da linguagem está correta. A prova disso são os milhares de linguagens de programação que usam esse conceito para executar um programa de computador, seja Fortran, Cobol, Lisp, C, C++, C#, Java, SQL, Ruby, Lisp, Prolog, Haskell, entre outros (Aho et al., 2007). Assim, o *framework* pode ser usado para avaliar automaticamente outras estruturas computacionais ou linguagens, como diagramas UML, código Java, Python, entre outros. Obviamente, cada linguagem segue a sua própria estrutura gramatical de construção que deve ser implementada, bem como existem formas diferentes de se medir esses objetos, e por isso são inseridas as métricas de engenharia de software no segundo nível.

No nível 2, as métricas de engenharia de software são usadas para medir um programa informático, seja SQL, DER, Python, JAVA, diagramas UML, entre outros (Pressman & Maxim, 2015). No LabDER, as métricas extraem informações de medidas dos componentes estruturais de um diagrama ER e de conceitos de BD, por exemplo, conceitos sobre relacionamentos, chave e domínio dos atributos.

As métricas são eficazes para extrair informações do software que pode ser medido objetivamente (Sommerville, 2016) e, por isso, são empregues na tarefa de quantificar as estruturas e conceitos de DER. As métricas extraídas das respostas do aluno e do professor formam o conjunto de dados de entrada dos algoritmos de aprendizagem computacional, que têm o objetivo de prever a avaliação do professor.

No nível 3, o objetivo da aprendizagem computacional é gerar o modelo preditivo e depois usá-lo para prever a nota do aluno. Para fazer isso, as métricas são importantes para fornecer informações aos algoritmos de aprendizagem computacional, que aprende e gera um modelo preditivo a partir de um conjunto de respostas dos alunos e professores e das avaliações de professores. Neste caso, trata-se de aprendizagem supervisionada (Russell & Norvig, 2009).

A primeira vantagem desta abordagem de avaliação automática é permitir múltiplas respostas corretas. A segunda vantagem é a previsão e classificação das notas de cada resposta

do aluno, que são geradas a partir do modelo preditivo que mede a distância entre a resposta do aluno e a resposta modelo através das métricas. A terceira vantagem é a qualidade do *feedback*, que pode ser descrita, no nível mais alto, em forma de conceitos de BD ou, no mais baixo nível, para detalhar os elementos de sintaxe que estão em falta ou excedem na resposta, auxiliando os alunos a identificar mais rapidamente o erro da solução submetida através de sugestões. A quarta vantagem dessa abordagem é oferecer ao aluno a possibilidade de refazer a solução. O aluno visualiza o *feedback* semântico com uma nota e decide se gostaria de criar uma nova resposta, que poderia ser ainda melhor do que a registrada no sistema.

4.3.1 Primeiro nível de avaliação - compiladores

4.3.1.1 Análise léxica

A análise léxica é o processamento de uma sequência de entrada de caracteres (como o código-fonte de um programa de computador) para produzir uma sequência de símbolos chamada “*tokens* lexicais” ou apenas *tokens* como saída (Singh et al., 2009). A análise léxica do *framework* recebe o DER como entrada e produz uma longa sequência de *tokens*. Um *token* pode ser composto por um único caráter ou uma sequência de caracteres, e podem ser classificados como sendo:

Identificador: O identificador é utilizado para referenciar os objetos de BD como tabelas, exibições, colunas, índices, gatilhos, procedimentos, restrições e regras. Por exemplo, a Figura 4-5 ilustra um diagrama ER e o comando SQL equivalente, o DER deve ser interpretado como uma instrução que define uma tabela com o identificador “TabelaX” e duas colunas com os identificadores “ColChave” e “Descricao”.

Palavras-chave reservadas: São palavras usadas para definir, manipular e acessar a BD. Essas palavras fazem parte da gramática da linguagem do SGBD, e são usadas para analisar e compreender as instruções e os lotes de SQL e por isso deve ser evitada a utilização dessas palavras para nomear objetos de BD. O DER e o seu SQL correspondente são ilustrados na Figura 4-5 e têm as seguintes palavra-chave: Create, Table, Integer, Primary Key e Varchar.

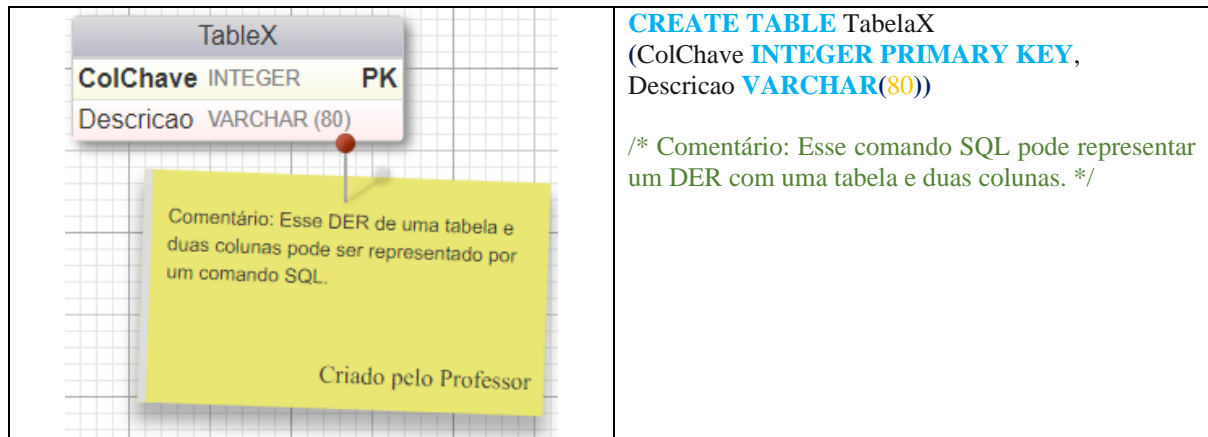


Figura 4-5. Exemplo de DER com uma tabela e duas colunas com código SQL correspondente.

Operadores: São símbolos que especificam uma ação que é executada numa ou mais expressões. Nessa categoria estão incluídos os operadores: aritméticos, lógicos, atribuição, resolução de escopo, bit a bit, conjunto, comparação, encadeamento da cadeia de caracteres, compostos e unários. Por exemplo, operadores matemáticos aritméticos (+, -, *, /, %), e operadores de comparação (>, <, >=, <=, <>, !=, !<, !>).

Separadores: São sinais de pontuação usados para agrupar sequências de *tokens* que têm um significado da unidade. Por exemplo: o separador ponto entre dois números inteiros, como 1.23, tem o significado de um número decimal, enquanto os dois pontos são usados como separador de horas, a barra invertida como separador de dia, mês e ano, o ponto e vírgula como separador de bloco de comando SQL, e os parênteses como separador de uma ou mais expressões e tipos especiais de dados.

Constantes: São valores constantes indicados diretamente, podem ser números (1-123, 3.14, 6.02), caráter 'a', e cadeia de caracteres 'algum texto'.

Comentários: São os símbolos /* abrem e os símbolos */ fecham um bloco de comentário na linguagem SQL, enquanto o comentário no DER é visual recorrendo a um *post-it* virtual, normalmente usados como um *feedback* adicional pelo professor (ver Figura 4-5).

Os *tokens* identificados nas respostas do professor e do aluno são armazenados na tabela símbolos na BD, de forma semelhante ao funcionamento de um compilador, servem de entrada para o analisador sintático e semântico (Singh et al., 2009). Os *tokens* permitidos no fluxo de entrada são descritos por expressões regulares. Cada expressão regular é associada a uma produção na gramática léxica que avalia os lexemas correspondentes à expressão regular. Novamente de forma análoga ao compilador, o trabalho do analisador léxico inclui (Su & Yan, 2011): a) produzir uma sequência de unidade atômica chamada *token*; b) remover os espaços em branco e os comentários da origem; c) acompanhar a localização dos *tokens*.

O analisador léxico consegue distinguir os *tokens* como palavras-chave reservadas, identificadores, operadores de relação, comentários, entre outros através de consultas entre a tabela de símbolo, dicionário de dados e outras consultas auxiliares. É verificada a exatidão da palavra de entrada, a grafia, a sua existência como objeto de BD. Pode fazer-se uso de expressões regulares para verificar se os *tokens* identificadores seguem um padrão de nomenclatura pré-definido ou, inclusive o recomendado nos livros de BD. Por exemplo, Coronel e Morris (2019) recomendam evitar o conceito de homónimos na nomenclatura de colunas com um carácter e um símbolo, como no caso da coluna “C_Name”, o “C” poderia ser interpretado como da tabela de Consultor ou Cliente. Uma expressão regular é capaz de identificar esse padrão e poderia fazer parte do *feedback* de identificadores. Bem como poderia existir uma expressão regular para detetar espaços em branco entre os caracteres, e assim por diante.

Em geral, o analisador léxico armazena os *tokens* na tabela da BD com um identificador incremental, isso auxilia no mapeamento do erro e sua localização como objeto. Por exemplo, o primeiro *token* é codificado como 1 e, o segundo é codificado como 2, e assim por diante. Na análise léxica há uma tarefa para verificar a declaração do identificador. A princípio isso implica que, para um identificador, é necessária apenas uma declaração, caso contrário, ela cometerá o erro da definição repetida. Não é permitido definir duas vezes um identificador, mesmo contendo tipos de dados diferentes. Não é permitido usar um identificador sem definição. Se esses regulamentos forem violados, o analisador léxico tratará como erro.

O programa principal do analisador léxico é constituído pelas seguintes partes: (a) a função principal que gere as entradas do DER, e (b) a função que inicia o analisador léxico com tokenizador. O tokenizador é usado para varrer o fluxo de entrada e obter o próximo *token* e colocar os dados na tabela da BD.

A Figura 4-6 ilustra o fluxo dos dados e o início do analisador léxico. O programa principal do analisador léxico invoca repetidamente o *tokenizer* e sub-rotinas selecionam o carácter de entrada atual para ver a que tipo de *token* que ele pertence, então a informação encontrada no *token* é armazenada ao mesmo tempo que o *token* é reconhecido. O *loop* é finalizado quando um *token* de fim do programa é reconhecido. O tratamento de erros é acionado no final do processamento do analisador léxico, para verificar a existência de *tokens* não reconhecidos pelo SGBD e emitir uma mensagem de erro personalizada.

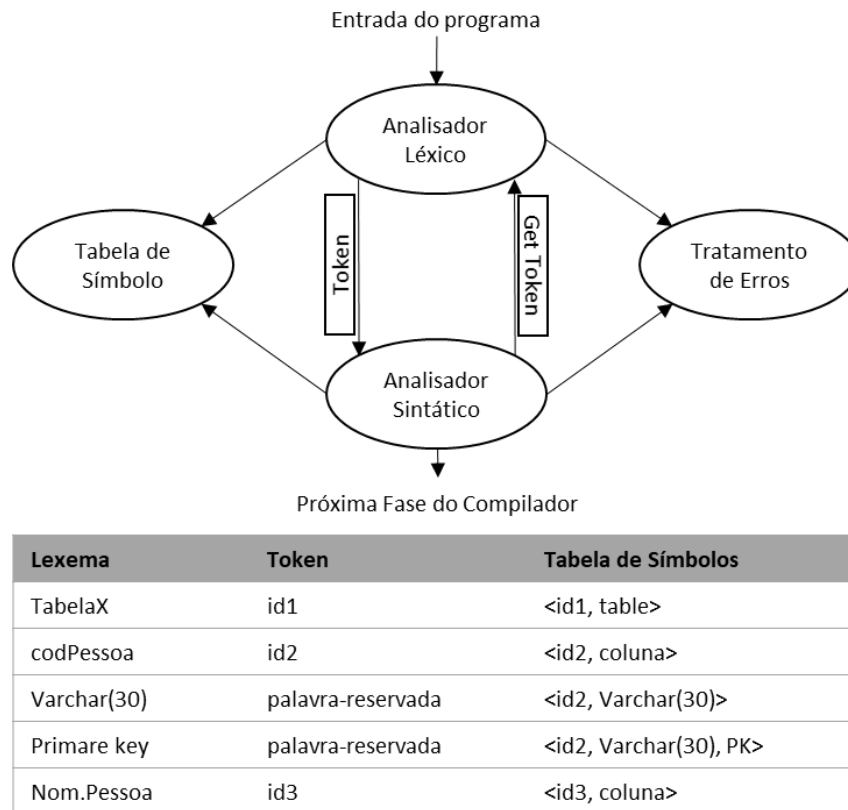


Figura 4-6. Fluxo do programa principal do analisador léxico.

Tratamento de erros

Os erros que o analisador léxico deteta podem ser agrupados nas seguintes classes (ver exemplo de erro Figura 4-7): 1) grafia errada das palavras reservadas, 2) erros de identificação, 3) erros numéricos, 4) erros de pontuação. Existem duas atitudes em relação aos erros. Uma delas segue estritamente as regras para tratar os erros: quando um erro é encontrado, então o *feedback* de erro é emitido ao utilizador com o tipo e posição do erro. A outra segue o aspeto da correção, que define as mensagens do *feedback*. No caso da correção, tem-se a preocupação de definir até que ponto se deve informar o utilizador dos elementos que estão em falta, sem que se forneça “uma resposta” ao aluno, este item é pormenorizado na análise semântica.

```

1  mysql> SELECT i FROM t INNER JOIN t AS t2;
2  ERROR 1052 (23000): Column 'i' in field list is ambiguous
  
```

Figura 4-7. Exemplo de erro - Identificador ambíguo.

Portanto, a análise léxica objetivamente verifica as palavras-chave reservadas, identificadores, constantes, operadores, separadores, e se os erros são causados por excesso ou falta de digitação. Por exemplo, para um número decimal, se tiver dois pontos decimais, obviamente, a correção pode ser feita eliminando-se um ponto decimal, mas é necessário

decidir qual deles deve ser excluído. Se o erro for causado por falta de carácter, também será necessário decidir qual o carácter que deve ser adicionado para torná-lo correto. Se o erro é causado porque há ocorrência do identificador sem definição, uma ocorrência de definição deve ser incluída. Mas, se para um identificador existem duas definições, e elas são inconsistentes, então uma delas deve ser excluída (ver Figura 4-8). No entanto, é necessário decidir cuidadosamente qual o identificador que deve ser mantido. Comparativamente, os tipos de erros na análise lexical não são muitos e a sua manipulação é bastante simples. A maioria dos erros é provocada por erros de digitação de um ou mais caracteres, falta ou excesso de digitação ou erros de ortografia.

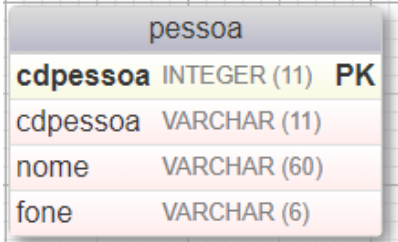
DER	SQL Equivalente
	<pre>CREATE TABLE pessoa (cdpessoa INT(11)NOT NULL, cdpessoa VARCHAR(11)NOT NULL, nome VARCHAR(60), fone VARCHAR(6), PRIMARY KEY (cdpessoa)) /* Comentário: O SQL representa o DER equivalente com uma tabela e quatro colunas. A coluna cdpessoa é declarada duas vezes. */</pre>
Mensagem de erro: Coluna duplicada	<pre>Error Code: 1060 Duplicate column name 'cdpessoa'</pre>

Figura 4-8. Exemplo de erro - Identificadores repetidos.

Ao se incorporar o tratamento de erros das linguagens existentes, torna-se o processo de verificação mais simples e ajuda a focar a construção de regras para componentes a serem avaliados sob o ponto de vista educacional. Ao incorporar o tratamento de erros do SGBD MySQL, temos a análise e deteção de 1.727 tratamento de erros, conforme consta no manual do catálogo de erros do SGBD (MySQL, 2018). Adicionalmente, ao integrar o catálogo de erros das linguagens, como no caso do SGBD, cria-se um ambiente mais realista pois, de facto, os alunos receberam um *feedback* nativo do SGBD, o que favorece uma aprendizagem prática.

A reutilização dos catálogos de erros permite uma análise sobre a frequência dos tipos de erros cometidos por aluno e associá-los ao percurso académico durante o uso da ferramenta, a fim de descobrir padrões de comportamento e propor melhorias nos ambientes educacionais. Por exemplo, a partir da análise de mais 500 mil respostas submetidas pelos alunos, podem-se inferir, as razões pelas quais o aluno comete um determinado erro em forma de regras, bem como dispor desta informação ao professor, para que possa direccionar as aulas, sem detrimento das ações dos alunos (A. Lino et al., 2019).

4.3.1.2 Análise sintática

Segundo Sebesta (2015), o analisador sintático é o coração do compilador, porque diversos outros componentes importantes, incluindo o analisador semântico e o gerador de código intermediário, são conduzidos pelas suas ações (Sebesta, 2015). No LabDER, o analisador sintático é o coração e o cérebro, porque, além de conduzir as ações entre os demais analisadores, tem a difícil tarefa de verificar se há uma equivalência mínima entre a resposta do professor e a resposta do aluno.

O objetivo da análise sintática é verificar se a sintaxe do programa de entrada está gramaticalmente correta, ou seja, validar se existe uma sequência válida de *tokens* gerados pelo analisador léxico, capaz de produzir expressões, sentenças e unidades de programas sem se preocupar com o significado (Mak, 2009). Uma gramática pode ser vista como um mecanismo para gerar frases ou elementos de uma linguagem, e é formada por um conjunto finito de regras com o objetivo de definir uma linguagem formal (Watson, 2017). Uma linguagem é o conjunto de todas as cadeias de *token* que são válidas de acordo com a gramática (Su & Yan, 2011). Assim, uma gramática define uma linguagem.

Uma sequência de *tokens* que é válida é uma declaração sintaticamente correta. Então, uma declaração é sintaticamente correta para a linguagem definida pela gramática se a gramática puder deduzir a declaração. Cada regra gramatical produz uma sequência de *tokens* válida da linguagem e, portanto, uma regra gramatical também é chamada de produção ou regra de produção. O conjunto de produções aplicadas sequencialmente a uma declaração sintaticamente correta é a derivação da instrução (Watson, 2017).

Portanto, a análise sintática do *framework* verifica uma equivalência sintática a partir da análise da resposta do professor, que identifica os conjuntos mínimos de produções equivalentes que devem estar presentes na resposta do aluno. Enquanto o interpretador do SGBD auxilia na tarefa de validar se as entradas são sintaticamente corretas ou se possuem erros, ou seja, valida se estão presentes todos *tokens*. Isto é, há uma dupla validação sintática, a primeira pelo SGBD que tem o propósito de identificar se a frase é sintaticamente correta, e a segunda pelo *framework* que tem o objetivo de encontrar a equivalência mínima entre duas respostas.

Segundo Jajodia et al. (1983), existem três requisitos que se devem cumprir para existir a equivalência entre dois DER. Assim, dois DER serão equivalentes se: i) a coleção total de conjuntos de entidades e valores descreve suficientemente todos os objetos que estão a ser modelados; ii) a coleção total de conjuntos de relacionamentos define relações sobre a coleção

total de conjuntos de entidades, e as entidades relacionadas são mapeadas por atributos para conjuntos de valores, em terceiro lugar; iii) a extensão dos domínios para os vários atributos (ou seja, seus conjuntos de valores) são os mesmos nos modelos que estão a ser considerados (Jajodia et al., 1983).

Para Barker (1992), existe equivalência entre dois DER quando: “dois DER destinados a modelar o mesmo mundo real são equivalentes se, e somente, as informações contidas num diagrama forem capturadas pelo segundo e vice-versa. Portanto, se as informações contidas num diagrama estiverem contidas no segundo, e o inverso também for verdadeiro, os diagramas serão equivalentes. A demonstração da equivalência exige a divisão em partes dos diagramas nos seus componentes e a análise de cada componente ou conjunto de componentes. Se ambos contiverem os mesmos componentes, no mesmo relacionamento, os diagramas serão equivalentes.” (Barker, 1992).

Seguindo as definições de Barker (1992) e Jajodia et al. (1983), o *framework* encontra a equivalência entre o DER do sistema e da resposta registada pelo aluno dividindo os componentes do DER nas suas partes constituintes, e indica, a partir das análises léxica, sintática e semântica, se o mínimo de componentes do DER da resposta registada no sistema estão presentes na resposta do aluno, e vice-versa.

O *feedback* é construído dinamicamente a partir da análise sintática que identifica os grupos de componentes que foram excedidos ou que estejam em falta em relação à resposta do professor. Então, o aluno recebe um *feedback* em forma de sugestões e uma lista de itens conceptuais que foram excedidos ou que estejam ausentes em sua resposta.

No geral, a fase de análise sintática do *framework* impõe as regras sintáticas que devem estar na resposta do aluno a partir da resposta do professor, e verifica então se essas regras foram cumpridas. Essa fase permite também o desenvolvimento de regras adicionais, independentemente da resposta do professor. Por exemplo, na definição de chave primária, para cada tabela deve existir pelo menos um campo chave primária (Coronel & Morris, 2019). Uma vez implementada essa regra, o *framework* verifica a existência dessa produção no DER, e emite um *feedback* de erro no caso de violação desta regra.

4.3.1.3 Análise semântica

A análise semântica da linguagem dá significado a uma declaração ou a uma expressão (Mak, 2009). Uma expressão pode estar sintaticamente correta e semanticamente inválida, pois a semântica avalia a expressão noutra perspetiva, ela identifica se as expressões estão a ser

aplicadas corretamente. As tarefas básicas desempenhadas por um compilador durante a análise semântica incluem a verificação de tipos, a verificação do fluxo, de controle e a verificação da singularidade da declaração de variáveis (Watson, 2017).

O MySQL faz a verificação de tipos de dados de forma diferente em projeções SQL e desenho de DER. Nas projeções são aplicadas as conversões automáticas dos tipos de dados, por exemplo, uma projeção com uma expressão de soma entre um número e uma coluna fornecem resultados diferentes dependendo do tipo de dado da coluna. Por outro lado, o tipo de dados é verificado na definição da BD, por exemplo, uma relação entre duas tabelas está correta se os tipos de dados das colunas do relacionamento forem exatamente iguais.

A análise semântica verifica o tipo de dados de cada *token* da expressão consultando a tabela de símbolo, que contém o catálogo de todos os *tokens* identificados pela análise léxica (ver Figura 4-9). Caso os tipos sejam diferentes, o tratamento de erros pode enviar um *feedback* para o caso de incompatibilidade de tipos de dados.

Lexema	Categoria	Domínio	PK	FK	Coluna FK	Tabela FK
Curso	<Tabela>					
Cur.Cod	<coluna>	inteiro	Sim			
Cur.Nome	<coluna>	Carácter	Não			
Pessoa.Cod	<coluna>	inteiro	Não	Sim	Pessoa.Cod	Pessoa

Figura 4-9. Exemplo de tabela de símbolo para verificação de tipo de dados do DER.

A verificação da singularidade da declaração verifica se há apenas uma única declaração para um determinado tipo de objeto de BD, qualquer duplicação é considerada como um erro semântico. Por exemplo, a API de design de DER permite criar duas relações entre duas tabelas usando os mesmos atributos, bem como definir duas tabelas com o mesmo nome, mesmo que essas definições estejam corretas do ponto de vista de sintaxe, estão erradas do ponto de vista semântico, seria redundante e recusado pelo SGBD. Portanto, o analisador semântico também deve identificar tabelas e relacionamentos duplicados.

O caso de singularidade da declaração é bastante usado pelo *framework* para identificar os conceitos de BD utilizados em excesso ou que estejam em falta na solução do aluno. A análise semântica funciona de forma análoga ao analisador sintático, pois também usa o SGBD para identificar alguns elementos de semântica enquanto usa a resposta do professor para mapear os conceitos de BD que devem ou não constar na resposta do aluno.

Na análise semântica, o *framework* mapeia todos os *tokens* da resposta do professor em forma de conceitos de BD, e identifica se esses conceitos estão presentes na resposta do aluno. Caso haja um excesso ou uma falta de conceitos de BD na resposta do aluno, o tratamento de erros gera uma mensagem de *feedback*. Por exemplo, “Excedeu o máximo de 2 relacionamentos ternários”, “Está em falta 1 relacionamento (M:N)”, “Excedeu a quantidade de tabelas”, “Quantidade de cardinalidade/especialização erradas na origem: 5”, entre outros.

O *framework* implementa as seguintes classes de conceitos de BD na análise semântica, são eles: tabela, colunas, domínio, chaves, campo opcional ou requerido, relacionamentos, força do relacionamento (fraco, forte), grau do relacionamento (unário, binário, ternário, n-Ary), cardinalidade/participação e especialização. Esses conceitos são quantificados na fase de métricas, e são usados no tratamento de erros para dar um *feedback* mais preciso sobre a quantidade de erros cometidos sobre um determinado conceito, igualmente para servir como entrada para a outra fase de aprendizagem computacional para prever a classificação final do aluno.

Sob o ponto de vista computacional, essa abordagem proporciona a identificação do erro exato e a implementação de diferentes estratégias de *feedback*. Assim, foi escolhida uma abordagem mais genérica para indicar quais os conceitos errados ao invés de um erro muito específico para alguns casos, e assim evitar a mesma limitação das abordagens anteriores de avaliação automática, onde os alunos conseguem adivinhar a resposta recorrendo a destaques visuais. Caso o *feedback* seja muito preciso, estaríamos apenas a alterar o problema de destaques visuais de lugar, que nas abordagens anteriores é originalmente encontrado no comando da questão.

Pode-se afirmar que a análise semântica do *framework* procura responder se as expressões dos conceitos de BD presentes na resposta do aluno fazem algum sentido quando comparadas com a resposta do professor. Por exemplo, independentemente de qualquer questão no sistema, mas dependente de uma resposta do professor, o *framework* é capaz de verificar se faz sentido a presença ou a ausência do conceito de especialização, bem como de qualquer outro conceito de BD abrangido nesta versão.

A tabela de símbolos é armazenada na BD do sistema, o que torna relativamente fácil incluir novas regras semânticas através de consultas SQL. Podem-se incluir regras para encontrar tabelas ou colunas duplicadas, bem como para detetar inversões de cardinalidades. Estes exemplos servem de novas regras a serem implementadas no *framework*. Antes de incluir qualquer regra, as mesmas devem ser cuidadosamente estudadas, uma vez que serão replicadas

para todos os utilizadores do sistema. A versão atual do *framework* cobre 100% dos conceitos de BD relacionais descritos no livro mais recente de Coronel e Morris (2019), e esses conceitos estão descritos na Tabela 2-2.

4.3.2 Segundo nível de avaliação - métricas

A métrica de Engenharia de Software é definida pela IEEE como “uma medida quantitativa do grau em que um sistema, componente ou processo possui um determinado atributo.” (IEEE Computer Society., 1990). Basili e Rombach (1988) defende que as métricas podem ser aplicadas para caracterizar, avaliar, prever e analisar código de um programa (Basili & Rombach, 1988). Basicamente, uma métrica ou um conjunto de métricas é um indicador que fornece informações sobre o processo, o produto ou o projeto (Pressman & Maxim, 2015).

As métricas têm vindo a ser amplamente utilizadas na área da informática para aferir o custo de sistemas (Jaiswal & Sharma, 2013), na alocação de recursos e performance (Pradipta, Aneesh, Dipankar, & Vaidyanathan, 2017), para estimar defeito de software (Ravi Kumar, Srinivasa Rao, & Bathini, 2019), nas diversas fases do ciclo de vida do desenvolvimento do software (Morrison, Moye, Pandita, & Williams, 2018), entre outros. As métricas são sobretudo uma abordagem quantitativa para medir a qualidade do *software*.

Pressman e Maxim (2015) apresentam desde as métricas mais elementares, como o número de linhas de código fonte, complexidade ciclomática de McCabe e o volume de Halstead, e outras métricas para aferir um processo ou projeto, tais como pontos por função, complexidade do dado e complexidade do sistema (Pressman & Maxim, 2015).

No contexto de BD, existem métricas para medir a qualidade da BD (Calero & Piattini, 2002), prever o esforço computacional de pequenas aplicações de BD relacionais (Bushra, Javed, Asma, & Shaista, 2010), medir a distância entre consultas em termos de processamento de entrada e saída (Braunmuller, Ester, Kriegel, & Sander, 2001), e por último e não menos importante, para entender e avaliar a complexidade estrutural dos diagramas ER (Eick, Lockemann, Eick, & Lockemann, 1985; Eick & Raupp, 1991; Genero et al., 2008; Kesh, 1995; D. L. Moody, Shanks, & Darke, 1998; Shepperd & Ince, 1991).

O segundo nível do processo de avaliação automática deve responder a como avaliar cada um dos elementos e conceitos do diagrama ER. Por outras palavras, deve fornecer indicadores fiáveis com a capacidade de medir a distância entre as respostas do aluno e do professor. Por se tratarem de métricas estruturais de um diagrama ER, segundo Genero et al.

(2008), todos os elementos de um diagrama ER contribuem para sua complexidade estrutural e são medidos conforme os diferentes tipos de modelo ER (Genero et al., 2008).

O trabalho de Genero et al. (2008) teve o objetivo de validar as métricas de DER na notação de Chen (1976) através de ensaios teóricos fundamentados na Teoria das Medidas proposta por (Poels & Dedene, 2000), o que significa que, do ponto de vista da teoria de medição, essas métricas realmente medem o atributo que pretendem medir. Ainda, nesse trabalho foi feita uma série de três ensaios relacionados, nos quais a relação entre os valores das métricas e as avaliações de especialistas foram analisadas estatisticamente quanto ao seu significado. Os resultados obtidos sugerem um conjunto de métricas estruturais que podem influenciar positivamente ou negativamente na concepção do modelo preditivo, a priori. A partir desse trabalho, sabe-se que o número de relacionamentos recursivos e número de especializações podem ter pouco ou nenhum efeito no modelo preditivo.

Encontrar formas de entender, avaliar e comparar quantitativamente DER são as metas deste nível de avaliação. A seguir, são identificados os elementos do DER que se considera como determinantes no processo de avaliação. Então, o conjunto de métricas é apresentado conjuntamente com as suas respectivas definições. As validações formais das métricas serão fornecidas na próxima seção, em conjunto com as definições formais utilizando construtores teóricos de conjuntos.

4.3.2.1 Identificação e seleção de métricas

O primeiro critério de seleção de métricas é a adaptação de um conjunto de métricas para se encaixar com os conceitos de BD, dando ênfase ao modelo *crow's foot*. Por outro lado, todos os elementos identificados do DER na seção de compiladores podem ser medidos, e isso pode ser uma mais-valia para o modelo preditivo, uma vez que tem como entrada as informações quantificáveis extraídas do DER. Assim, um segundo critério de seleção e identificação de métricas é desenvolver, converter ou derivar um conjunto de métricas que estejam associadas aos construtores oferecidos pela API que implementa o modelo *crow's foot*.

As métricas utilizadas nesse projeto de investigação são inspiradas na proposta de (Genero et al., 2008), assim, foram convertidas algumas métricas e desenvolvidas novas métricas, uma vez que o trabalho de Genero et al. (2008) está centrado no modelo ER de (P. P.-S. Chen & Pin-Shan, 1976). As seguintes métricas foram então convertidas: entidades em tabelas e atributos em colunas. Algumas métricas são específicas do modelo de Peter Chen. Por exemplo, os atributos multi-valor, atributos simples e compostos, e atributos derivados são implementados de forma diferente no *crow's foot* e por isso foram descartados. Por outro lado,

os conceitos de restrições de campo opcional e o domínio do atributo estão implementadas no modelo *crow's foot*, disponível na API selecionada para este projeto. As métricas selecionadas são apresentadas com sigla e descrição do objetivo que se propõe avaliar e o conceito de BD envolvido.

As métricas foram mapeadas para quantificar os conceitos relacionados à BD. Em resumo, os principais conceitos que se pretendem medir são: tabela, colunas, colunas requeridas, colunas opcionais, domínio da coluna, chave primária, chave primária composta, relacionamento, relacionamento fraco, relacionamento forte, grau do relacionamento, cardinalidade, participação, especialização/generalização (ver a descrição em Tabela 2-2). A quantificação de cada um desses conceitos BD, torna o *feedback* semântico mais preciso.

A Tabela 4-1 descreve as métricas aplicadas no diagrama, cada uma focada num conceito diferente do DER. A avaliação da complexidade estrutural do DER é determinada pelos diferentes elementos que o compõem, como entidades, atributos, relacionamentos, entre outros. Esta avaliação está em consonância com a teoria de complexidade de Genero et al. (2008), “fundamentada no número de (diferentes tipos de) elementos e no número de (diferentes tipos de) relações (dinamicamente variáveis) entre eles.” (Genero et al., 2008).

Tabela 4-1. Métricas para medir o diagrama de entidade e relacionamento no LabDER.

Métrica	Descrição	Conceitos
<i>NT</i>	A métrica Número de Tabelas é definida como o número de tabelas dentro de um diagrama ER	Tabelas
<i>NC</i>	A métrica Número de Colunas é definida como o número total de colunas definidos no DER. Nessa métrica, todas as colunas são incluídas.	Colunas
<i>NCOpc</i>	A métrica Número de Colunas Opcionais é definida como o número de colunas opcionais no DER.	Coluna Requerido
<i>NCRqr</i>	A métrica Número de Colunas Requerido é definida como o número de colunas requeridas no DER.	Coluna Opcional
<i>NCNum</i>	A métrica Número de Colunas do Domínio Numérico é definida como o número de colunas do tipo numérico, elas podem ser ('Integer', 'Tinyint', 'Smallint', 'Mediumint', 'Int', 'Bigint', 'Decimal', 'Float', 'Double') dentro do DER. O valor de NCNum é sempre menor ou igual ao valor de NC.	Domínio
<i>NCCrc</i>	A métrica Número de Colunas do Domínio Caráter é definida como o número de colunas do tipo Caráter, elas podem ser ('Char', 'Varchar', 'Mediumtext', 'Binary', 'Varbinary', 'Blob') dentro do DER. O valor de NCCrc é sempre estritamente menor ou igual ao valor de NC.	Domínio
<i>NCDtt</i>	A métrica Número de Colunas do Domínio Data Hora é definida como o número de colunas do tipo data hora, elas podem ser ('Date', 'Time', 'Datetime', 'Year', 'Timestamp') dentro do DER. O valor de NCDtt é sempre menor ou igual ao valor de NC.	Domínio
<i>NCMsc</i>	A métrica Número de Colunas do Domínio Miscelânea é definida como o número de colunas do tipo miscelâneas, elas podem ser ('ENUM', 'SET', 'Bit') dentro do DER. O valor de NCMsc é sempre menor ou igual ao valor de NC.	Domínio
<i>NCPk</i>	A métrica Número de Colunas Chaves Primária é definida como o número de colunas chaves primárias definidos no DER.	Chave Primária
<i>NR</i>	A métrica Número de Relacionamentos é definida como o número total de relacionamentos no DER.	Relacionamento
<i>NR:1N</i>	A métrica Número de Relacionamentos M:N é definida como o número de relacionamentos M:N no DER. O valor de M:N é menor ou igual ao valor de NR.	Relacionamento
<i>NR:MN</i>	A métrica Número de Relacionamentos 1:N é definida como o número de relacionamentos 1:N no DER. O valor de 1:N é menor ou igual ao valor de NR.	Relacionamento
<i>NRUna</i>	A métrica Número de Relacionamentos Unários é definida como o número de relacionamentos unários no DER. O valor é menor ou igual ao valor de NR. Todo relacionamento unário é recursivo.	Grau do relacionamento
<i>NRBin</i>	A métrica Número de Relacionamentos Binários é definida como o número de relacionamentos binários no DER. O valor é menor ou igual ao valor de NR.	Grau do relacionamento
<i>NRTer</i>	A métrica Número de Relacionamentos Ternários é definida como o número de relacionamentos ternários no DER. O valor é menor ou igual ao valor de NR.	Grau do relacionamento
<i>NRNAry</i>	A métrica Número de Relacionamentos N-Ary é definida como o número de relacionamentos com grau maior ou igual a quatro relacionamentos dentro de um diagrama ER. O valor é menor ou igual ao valor de NR.	Grau do relacionamento
<i>NREsp</i>	A métrica Número de Relacionamentos especializadas é definida como o número de relacionamentos do tipo especialização/generalização no DER. O valor é menor ou igual ao valor de NR.	Especialização/ generalização
<i>NRFor</i>	A métrica Número de Relacionamentos Forte é definida como o número de relacionamentos forte no DER. O valor é menor ou igual ao valor de NR.	Força do relacionamento
<i>NRFra</i>	A métrica Número de Relacionamentos Fraco é definida como o número de relacionamentos fraco no DER. O valor é menor ou igual ao valor de NR.	Força do relacionamento
<i>NRCrdPk11</i>	A métrica Número de Cardinalidade/Participação na Chave Primária do Relacionamentos é definida como o número do tipo de cardinalidade/participação (um e somente um) na chave primária do relacionamento no DER.	Cardinalidade/ Participação
<i>NRCrdPk01</i>	A métrica Número de Cardinalidade/Participação na Chave Primária do Relacionamentos é definida como o número do tipo de cardinalidade/participação (zero ou um) na chave primária do relacionamento no DER.	Cardinalidade/ Participação
<i>NRCrdPk1N</i>	A métrica Número de Cardinalidade/Participação na Chave Primária do Relacionamentos é definida como o número do tipo de cardinalidade/participação (um ou muitos) na chave primária do relacionamento no DER.	Cardinalidade/ Participação
<i>NRCrdPk0N</i>	A métrica Número de Cardinalidade/Participação na Chave Primária do Relacionamentos é definida como o número do tipo de cardinalidade/participação (zero ou muitos) na chave primária do relacionamento no DER.	Cardinalidade/ Participação
<i>NRCrdFk11</i>	A métrica Número de Cardinalidade/Participação na Chave Estrangeira do Relacionamentos é definida como o número do tipo de cardinalidade/participação (um e somente um) na chave estrangeira do relacionamento no DER.	Cardinalidade/ Participação
<i>NRCrdFk01</i>	A métrica Número de Cardinalidade/Participação na Chave Estrangeira do Relacionamentos é definida como o número do tipo de cardinalidade/participação (zero ou um) na chave estrangeira do relacionamento no DER.	Cardinalidade/ Participação
<i>NRCrdFk1N</i>	A métrica Número de Cardinalidade/Participação na Chave Estrangeira do Relacionamentos é definida como o número do tipo de cardinalidade/participação (um ou muitos) na chave estrangeira do relacionamento no DER.	Cardinalidade/ Participação
<i>NRCrdFk0N</i>	A métrica Número de Cardinalidade/Participação na Chave Estrangeira do Relacionamentos é definida como o número do tipo de cardinalidade/participação (zero ou muitos) na chave estrangeira do relacionamento no DER.	Cardinalidade/ Participação

A seguir, as métricas são definidas informalmente em linguagem natural (ver Tabela 4-1), e são definidas formalmente na secção de validação. Para exemplificar cada uma dessas definições, as métricas são aplicadas ao DER ilustrado na Figura 4-10, exemplo extraído de Coronel e Morris (2019).

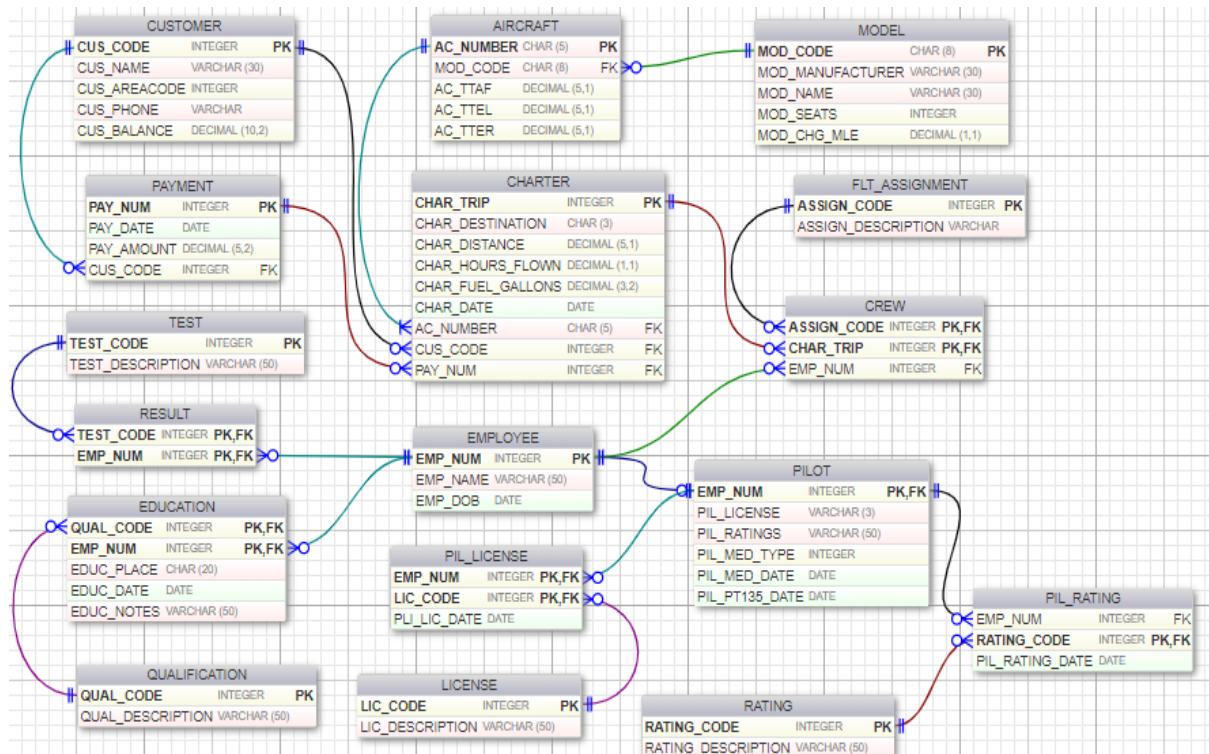


Figura 4-10. Exemplo de um diagrama de entidade e relacionamento (Coronel & Morris, 2019).

O exemplo da aplicação das métricas na Figura 4-10 está descrito na Tabela 4-2. Neste caso, a métrica número de tabelas (NT) é 17, o número de colunas (NC) é 63, número de colunas chaves primárias (NCPk) é 22, o número de relacionamento (NR) é 17, as demais listadas na Tabela 4-2.

Tabela 4-2. Valores métricos para o diagrama ER da figura anterior.

Métrica	Valores	Conceito	Descrição
<i>NT</i>	17	Tabelas	Total de Tabelas
<i>NC</i>	63	Colunas	Total de Colunas
<i>NCOpc</i>	62	Coluna Opcional	Número de Colunas Opcionais
<i>NCRqr</i>	1	Coluna Requerido	Número de Colunas Requeridas
<i>NCNum</i>	36	Domínio	Número de Colunas do Domínio Numérico
<i>NCCrc</i>	19	Domínio	Número de Colunas do Domínio Carácter
<i>NCDtt</i>	8	Domínio	Número de Colunas do Domínio Data Hora
<i>NCMsc</i>	0	Domínio	Número de Colunas do Domínio Miscelânea
<i>NCPk</i>	22	Chave Primária	Número de Chaves Primária
<i>NR</i>	17	Relacionamento	Total de Relacionamentos
<i>NR:IN</i>	16	Relacionamento	Número de Relacionamentos 1:N
<i>NR:MN</i>	6	Relacionamento	Número de Relacionamentos M:N
<i>NRUna</i>	0	Grau do relacionamento	Número de Relacionamentos de Grau Unário
<i>NRBin</i>	4	Grau do relacionamento	Número de Relacionamentos de Grau Binário
<i>NRTer</i>	2	Grau do relacionamento	Número de Relacionamentos de Grau Ternário
<i>NRNAry</i>	0	Grau do relacionamento	Número de Relacionamentos de Grau NÁry
<i>NREsp</i>	1	Especialização/ generalização	Número de Relacionamentos de Especialização
<i>NRFor</i>	11	Força do relacionamento	Número de Relacionamentos de Fortes
<i>NRFra</i>	6	Força do relacionamento	Número de Relacionamentos de Fracos
<i>NRCrdPk11</i>	17	Cardinalidade/ Participação	Número de Relacionamentos do tipo 1:1 no PK
<i>NRCrdPk01</i>	0	Cardinalidade/ Participação	Número de Relacionamentos do tipo 0:1 no PK
<i>NRCrdPk1N</i>	0	Cardinalidade/ Participação	Número de Relacionamentos do tipo 1:N no PK
<i>NRCrdPk0N</i>	0	Cardinalidade/ Participação	Número de Relacionamentos do tipo 0:N no PK
<i>NRCrdFk11</i>	0	Cardinalidade/ Participação	Número de Relacionamentos do tipo 1:1 no FK
<i>NRCrdFk01</i>	1	Cardinalidade/ Participação	Número de Relacionamentos do tipo 0:1 no FK
<i>NRCrdFk1N</i>	1	Cardinalidade/ Participação	Número de Relacionamentos do tipo 1:N no FK
<i>NRCrdFk0N</i>	15	Cardinalidade/ Participação	Número de Relacionamentos do tipo 0:N no FK

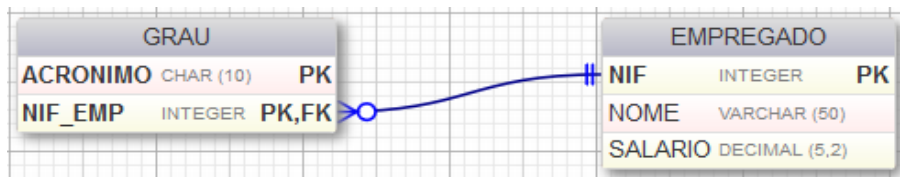
4.3.2.2 Validação das métricas

A validação das métricas previamente identificadas foi baseada no *framework* denominado Distância (Poels & Dedene, 2000), que foi utilizado por Genero et al. (2008) para validar as métricas do DER na notação do Peter Chen (1976). A validação é apresentada em três partes: a) a primeira parte descreve como todos os elementos do DER podem ser modelados por abstrações de medida que tomam a forma de conjuntos; b) a segunda parte visa definir distâncias entre abstrações de medida com uma solução genérica de construção de medidas de DER; e c) a terceira parte visa construir as medidas, ou seja, quantificar distâncias entre abstrações de medida (Poels & Dedene, 2000). Uma vez que a solução consiste em definições genéricas de medida de distância, então as medidas podem ser exemplos de qualquer tipo de conjunto. Como exemplo ilustrativo, o *framework* Distância foi aplicado na validação da métrica de Número de Colunas (NC). Todas as demais métricas foram validadas de forma análoga, e, por conseguinte, apenas as suas definições são apresentadas.

Validação da métrica de NC baseada na Distância:

A métrica de NC foi definida como o número de colunas definidas dentro do DER (ver Tabela 4-1). Essa medida deve atestar a complexidade estrutural do DER sob a perspectiva de colunas dentro do DER, sendo a complexidade sempre maior quando tiver mais colunas dentro do DER e menor quando tiver menos colunas dentro do DER. A validação da métrica NC tem como referência os diagramas ilustrados na Figura 4-11.

(Diagrama A)



(Diagrama B)

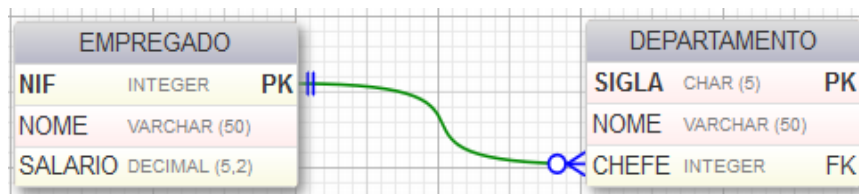


Figura 4-11. Dois exemplos de diagramas ER.

1ª Definição de abstração de medida:

O Universo de DER (notação: $UDER$) é o conjunto de elementos que são caracterizados por uma propriedade de complexidade estrutural (referido como o conjunto P), isto é, o conjunto de todos os DER sintaticamente corretos concebíveis que são relevantes para algum Universo do Discurso (UoD). A métrica de NC (referida como pty) representa a propriedade de complexidade estrutural do número de colunas definidos dentro de um elemento de $UDER$.

Seja o UC o Universo de Colunas relevante para o UoD . Então o conjunto de colunas definidos dentro de um $DER \in UDER$, denotado por $SA(DER)$, é um subconjunto de UC . Os conjuntos de colunas definidos nos DER de $UDER$ são elementos do domínio do conjunto de C (notação: $\wp(UC)$). Deste modo, pode-se equacionar o conjunto de abstrações de medição para $\wp(UC)$ e definir a função de mapeamento como:

$$abs_{nc}: UDER \rightarrow \wp(UC): DER \rightarrow SA(DER)$$

Essa função mapeia o DER no seu conjunto de colunas, que é uma representação do DER para o aspeto da complexidade estrutural da métrica NC. Aplicada esta função ao exemplo da Figura 4-11 temos:

$$abs_{nc}(DER A) = SA(DER A) = \{Grau.Acronimo, Grau.Nif, Empregado.Nif, Empregado.Nome, Empregado.Salario\}$$

$$abs_{nc}(DER B) = SA(DER B)$$

$$= \{Empregado.Nif, Empregado.Nome, Empregado.Salario, Departamento.Sigla, Departamento.Nome, Departamento.Chefe\}$$

2ª Definição de distâncias entre abstrações de medida:

Um conjunto de funções elementares de transformação para $\wp(UC)$, denotado por $T_{e-\wp}(UC)$, deve ser encontrado para que qualquer conjunto de colunas possa ser transformado em qualquer outro conjunto de colunas. Segundo Poels e Dedene (2000), duas funções de transformação elementares são necessárias para transformar conjuntos: uma que adiciona um elemento a um conjunto e outra que remove um elemento a um conjunto. Assim, dados dois conjuntos de colunas $s_1 \in \wp(UC)$ e $s_2 \in \wp(UC)$, s_1 pode ser transformado em s_2 removendo primeiro todas as colunas de s_1 que não estão em s_2 e, em seguida, adicionando todas as colunas em s_1 que estão em s_2 , mas não estavam originalmente em s_1 . No pior cenário, s_1 deve ser transformado em s_2 recorrendo a um conjunto vazio de colunas. Formalmente, $T_{e-\wp}(UC) = \{t_{0-\wp}(UC), t_{1-\wp}(UC)\}$, onde $t_{0-\wp}(UC)$ e $t_{1-\wp}(UC)$ são definidos como:

$$t_{0-\wp}(UC): \wp(UC) \rightarrow UC(UC): s \rightarrow s \cup \{c\}, com \in UC$$

$$t_{1-\wp}(UC): \wp(UC) \rightarrow UC(UC): s \rightarrow s - \{c\}, com \in UC$$

Então, a distância entre $SA(DER A)$ e $SA(DER B)$ é definida por uma sequência de transformações elementares que remove Grau.Acronimo e Grau.Nif de $SA(DER A)$ e que adiciona Departamento.Sigla, Departamento.Nome, e Departamento.Chefe para $SA(DER A)$. Sendo o suficiente uma sequência de cinco transformações elementares para transformar $SA(DER A)$ em $SA(DER B)$.

3ª Quantificação de distâncias entre abstrações de medida:

Segundo Poels e Dedene (2000), o modelo de diferença simétrica pode ser usado para definir uma métrica para distâncias entre conjuntos. Assim, a métrica $\delta_{\wp}(UC)$ pode ser definida como:

$$\delta_{\wp}(UC): \wp(UC) \times \wp(UC) \rightarrow \mathbb{R}: (s, s') \rightarrow |s - s'| + |s' - s|$$

Essa definição afirma que a distância entre dois conjuntos de colunas é medida da contagem da menor sequência de transformações elementares entre esses conjuntos. Note que, para qualquer elemento em s , mas não em s' e para qualquer elemento em s' , mas não em s , é necessária uma transformação elementar (Poels & Dedene, 2000). O modelo de diferença simétrica resultou num valor de 5 para a distância entre os conjuntos de colunas do $DER A$ e $DER B$. Formalmente temos:

$$\begin{aligned}
& \delta_{\emptyset}(UA)(SA(DER A), SA(DER B)) \\
&= |\{Grau. Acronimo, Grau. Nif, Empregado. Nif, Empregado. Nome, Empregado. Salario\}| \\
&- |\{Empregado. Nif, Empregado. Nome, Empregado. Salario, Departamento. Sigla, Departamento. Nome, Departamento. Chefe\}| \\
&+ |\{Empregado. Nif, Empregado. Nome, Empregado. Salario, Departamento. Sigla, Departamento. Nome, Departamento. Chefe\}| \\
&- |\{Grau. Acronimo, Grau. Nif, Empregado. Nif, Empregado. Nome, Empregado. Salario\}| \\
&= |\{Grau. Acronimo, Grau. Nif\}| + |\{Departamento. Sigla, Departamento. Nome, Departamento. Chefe\}| = 5
\end{aligned}$$

É necessária uma abstração de referência para a propriedade de complexidade estrutural, e o ponto de referência para medição é o conjunto vazio de atributos. Pode-se argumentar que um DER sem atributos definidos possui menor complexidade estrutural. Então a função de mapeamento é definida como:

$$ref_{NC}: UDER \rightarrow \wp(UC): DER \rightarrow \emptyset$$

A medida para a propriedade da métrica de NC pode ser formalmente definida como uma função que devolve a qualquer $DER \in UDER$ o valor da métrica $\delta_{\wp(UC)}$ a par de conjuntos $SC(DER)$ e \emptyset :

$$\begin{aligned}
\forall DER \in UDER: NC(DER) &= \delta_{\wp(UC)}(SC(DER), \emptyset) \\
&= |SC(DER) - \emptyset| + |\emptyset - SC(DER)| \\
&= |SC(ERD)|
\end{aligned}$$

Como resultado, uma métrica que devolve o número de colunas num DER qualifica-se como uma medida da propriedade de complexidade estrutural que é determinada pela quantidade de colunas definidas dentro de um DER.

Validação baseada em distância das outras métricas de complexidade estrutural

Todas essas propriedades de complexidade estrutural podem ser modeladas por uma abstração de conjunto e todas as abstrações de referência assumem a forma de um conjunto vazio. Como consequência, todas as métricas podem ser validadas como medidas que contam os elementos na abstração do conjunto. As abstrações de medição, que foram usadas ao aplicar DISTÂNCIA para as métricas de complexidade estrutural do diagrama ER, são listadas na Tabela 4-3.

Tabela 4-3. Abstrações de medida para as métricas de complexidade estrutural do DER.

Métrica	Abstração de Medida
<i>NT</i>	$abs_{NT}: UDER \rightarrow \wp(UT): DER \rightarrow ST(DER)$ onde UT é o Universo de Tabelas relevante do $UoD, ST(DER) \subseteq UT$ é o conjunto de tabelas dentro de um DER.
<i>NC</i>	$abs_{NC}: UDER \rightarrow \wp(UC): DER \rightarrow SC(DER)$ onde UC é o Universo de Colunas relevante do $UoD, SC(DER) \subseteq UC$ é o conjunto de colunas dentro de um DER.
<i>NCOpc</i>	$abs_{NCOpc}: UDER \rightarrow \wp(UC): DER \rightarrow SCOpc(DER)$ em que UC é o Universo de Colunas relevante do $UoD, SCOpc(DER) \subseteq UC$ é o conjunto de colunas opcionais dentro do DER.
<i>NCRqr</i>	$abs_{NCRqr}: UDER \rightarrow \wp(UC): DER \rightarrow SCRqr(DER)$ em que UC é o Universo de Colunas relevante do $UoD, SCRqr(DER) \subseteq UC$ é o conjunto de colunas requeridas dentro do DER.
<i>NCNum</i>	$abs_{NCNum}: UDER \rightarrow \wp(UC): DER \rightarrow SCNum(DER)$ em que UC é o Universo de Colunas relevante do $UoD, SCNum(DER) \subseteq UC$ é o conjunto de colunas do domínio numérico dentro do DER.
<i>NCCrc</i>	$abs_{NCCrc}: UDER \rightarrow \wp(UC): DER \rightarrow SCCrc(DER)$ em que UC é o Universo de Colunas relevante do $UoD, SCCrc(DER) \subseteq UC$ é o conjunto de colunas do domínio caráter dentro do DER.
<i>NCDtt</i>	$abs_{NCDtt}: UDER \rightarrow \wp(UC): DER \rightarrow SCDtt(DER)$ em que UC é o Universo de Colunas relevante do $UoD, SCDtt(DER) \subseteq UC$ é o conjunto de colunas do domínio data hora dentro do DER.
<i>NCMsc</i>	$abs_{NCMsc}: UDER \rightarrow \wp(UC): DER \rightarrow SCMsc(DER)$ em que UC é o Universo de Colunas relevante do $UoD, SCMsc(DER) \subseteq UC$ é o conjunto de colunas do domínio miscelâneas dentro do DER.
<i>NCPk</i>	$abs_{NCPk}: UDER \rightarrow \wp(UC): DER \rightarrow SCPk(DER)$ em que UC é o Universo de Colunas relevante do $UoD, SCPk(DER) \subseteq UC$ é o conjunto de colunas chave primária dentro do DER.
<i>NR</i>	$abs_{NR}: UDER \rightarrow \wp(UR): DER \rightarrow SR(DER)$ onde UR é o Universo de Relacionamentos relevante para um $UoD, SR(DER) \subseteq UR$ é o conjunto de relacionamentos dentro de um DER.
<i>NR:1N</i>	$abs_{NR:1N}: UDER \rightarrow \wp(UR): DER \rightarrow SR:1N(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SR:1N(DER) \subseteq UR$ é o conjunto de relacionamentos 1:N dentro do DER.
<i>NR:MN</i>	$abs_{NR:MN}: UDER \rightarrow \wp(UR): DER \rightarrow SR:MN(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SR:MN(DER) \subseteq UR$ é o conjunto de relações M:N dentro do DER.
<i>NRUna</i>	$abs_{NRUna}: UDER \rightarrow \wp(UR): DER \rightarrow SRUna(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRUna(DER) \subseteq UR$ é o conjunto de relacionamentos unários dentro do DER.
<i>NRBin</i>	$abs_{NRBin}: UDER \rightarrow \wp(UR): DER \rightarrow SRBin(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRBin(DER) \subseteq UR$ é o conjunto de relacionamentos binários dentro do DER.
<i>NRTer</i>	$abs_{NRTer}: UDER \rightarrow \wp(UR): DER \rightarrow SRTer(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRTer(DER) \subseteq UR$ é o conjunto de relacionamentos ternários dentro do DER.
<i>NRNAry</i>	$abs_{NRNAry}: UDER \rightarrow \wp(UR): DER \rightarrow SRNAryR(DER)$ onde UR é o Universo de Relacionamentos relevantes para o $UoD, SRNAry(DER) \subseteq UR$ é o conjunto de relacionamentos NAry dentro do DER.
<i>NREsp</i>	$abs_{NREsp}: UDER \rightarrow \wp(UR): DER \rightarrow SREsp(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SREsp(DER) \subseteq UR$ é o conjunto de relacionamentos de especialização dentro do DER.
<i>NRFor</i>	$abs_{NRFor}: UDER \rightarrow \wp(UR): DER \rightarrow SRFor(DER)$ onde UR é o Universo de Relacionamentos relevantes para um $UoD, SRFor(DER) \subseteq UR$ é o conjunto de relacionamentos forte dentro de um DER.
<i>NRFra</i>	$abs_{NRFra}: UDER \rightarrow \wp(UR): DER \rightarrow SRFra(DER)$ onde UR é o Universo de Relacionamentos relevantes para um $UoD, SRFra(DER) \subseteq UR$ é o conjunto de relacionamentos fraco dentro de um DER.
<i>NRCrdPk11</i>	$abs_{NRCrdPk11}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdPk11(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdPk11(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 1:1 na chave primária dentro do DER.
<i>NRCrdPk01</i>	$abs_{NRCrdPk01}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdPk01(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdPk01(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 0:1 na chave primária dentro do DER.
<i>NRCrdPk1N</i>	$abs_{NRCrdPk1N}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdPk1N(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdPk1N(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 1:N na chave primária dentro do DER.
<i>NRCrdPk0N</i>	$abs_{NRCrdPk0N}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdPk0N(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdPk0N(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 0:N na chave primária dentro do DER.
<i>NRCrdFk11</i>	$abs_{NRCrdFk11}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdFk11(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdFk11(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 1:1 na chave estrangeira dentro do DER.
<i>NRCrdFk01</i>	$abs_{NRCrdFk01}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdFk01(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdFk01(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 0:1 na chave estrangeira dentro do DER.
<i>NRCrdFk1N</i>	$abs_{NRCrdFk1N}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdFk1N(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdFk1N(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 1:N na chave estrangeira dentro do DER.
<i>NRCrdFk0N</i>	$abs_{NRCrdFk0N}: UDER \rightarrow \wp(UR): DER \rightarrow SRCrdFk0N(DER)$ onde UR é o Universo de Relacionamentos relevantes do $UoD, SRCrdFk0N(DER) \subseteq UR \cap UC$ é o conjunto de relacionamentos com cardinalidade 0:N na chave estrangeira dentro do DER.

O resultado da aplicação das métricas durante o processo de avaliação automática dos DER é utilizado para aumentar a precisão do *feedback* semântico, fornecendo uma medida quantitativa entre a diferença das estruturas do DER do aluno e do professor. Desta forma, o

feedback envia uma mensagem indicando se excedeu ou se está em falta um conceito de BD e a respetiva unidade de medida da distância.

Devido à quantidade elevada de métricas, que se pretende medir, existe um risco de poluir a interface do *feedback*, e por isso pensámos em agrupar pelo menos as métricas de cardinalidade. Então, no caso de erros, ao invés de apresentar oito métricas de cardinalidades, apresentamos duas medidas que representam o agrupamento dessas métricas em categorias de chaves estrangeiras e chaves primárias. Desta forma, o *feedback* apresenta o total de erros em cada grupo, por exemplo, estão erradas 3 cardinalidades na origem dos relacionamentos (PK), e/ou estão erradas 5 cardinalidades no destino do relacionamento (FK).

As métricas extraídas dos DER de alunos e professores, são a entrada da aprendizagem computacional, relatada no terceiro nível, na próxima secção.

4.3.3 Terceiro nível de avaliação - aprendizagem computacional

Esta secção descreve o papel da aprendizagem computacional no processo de avaliação automática de DER. São também fornecidos detalhes de como as entradas trabalham em conjunto para gerar a saída, isto é, a classificação final da resposta do aluno.

A complexidade de avaliar automaticamente o DER está na dificuldade em encontrar um método computacional capaz de explicar, numa escala específica de avaliação, a relação entre as distâncias das métricas do DER do aluno e da resposta modelo. As 27 métricas (ver Tabela 4-3) representam todo o conjunto de elementos que se pretende avaliar, sendo cada uma das métricas o conjunto de elementos básicos das estruturas do DER. Caso seja feita uma transformação de distância de mínimo e máximo (numa escala de 0 a 1) que exprima a relação entre os DER, então poderíamos utilizar pelo menos 11 elementos para representar a distância em percentagem, por exemplo (1, 0.9, 0.8, ..., 0.1, 0), onde 1 representa “totalmente diferente” e 0 “totalmente equivalente”. Neste caso, uma combinação simples de 27 elementos distintos agrupados em 10 resultaria em 8.436.285 combinações possíveis, mostrando-se inviável implementar todas essas combinações através de códigos paramétricos.

O elevado número de variáveis envolvidas na avaliação de DER limita a aplicação de outras abordagens computacionais para resolução deste problema, por exemplo lógica difusa, cujo tamanho da base de regras depende do número de variáveis linguísticas do sistema. Caso fossem cinco variáveis linguísticas (falhou, fraco, suficiente, bom, excelente) para cada um dos 27 conjuntos de entrada, então o tamanho da base de regras seria de 14.348.907 regras.

Outro desafio encontrado neste processo é o de definir um sistema de avaliação que aceite os diferentes estilos de avaliação dos professores. A complexidade aumenta em detrimento da subjetividade intrínseca ao processo de avaliação e aos diferentes critérios que cada professor possa considerar durante a avaliação. Portanto, é um desafio conceber um sistema que seja aceite pela comunidade docente na área de bases de dados.

A solução por algoritmos de aprendizagem computacional é recomendável para ser aplicada no contexto do LabDER pela sua grande capacidade de generalização, e quando aplicada nesse contexto educacional irá obter um modelo preditivo com as características mais comuns empregadas pelos professores durante o processo de avaliação.

4.3.3.1 O papel da aprendizagem computacional no *framework*

O objetivo principal deste terceiro nível de avaliação é usar um modelo preditivo para prever a classificação final da resposta do aluno e servir como *feedback* da avaliação (ver Figura 4-12). O modelo preditivo é concebido recorrendo a algoritmos de aprendizagem computacional, que tem como entrada as métricas de DER extraídas da resposta do aluno e da resposta modelo, bem como as notas da avaliação do professor para cada DER. Como procedimento comum nas atividades de geração de aprendizagem computacional, as métricas são transformadas numa escala entre 0 e 1, através do cálculo de distância de mínimo e máximo, sendo zero totalmente iguais e um totalmente diferentes.

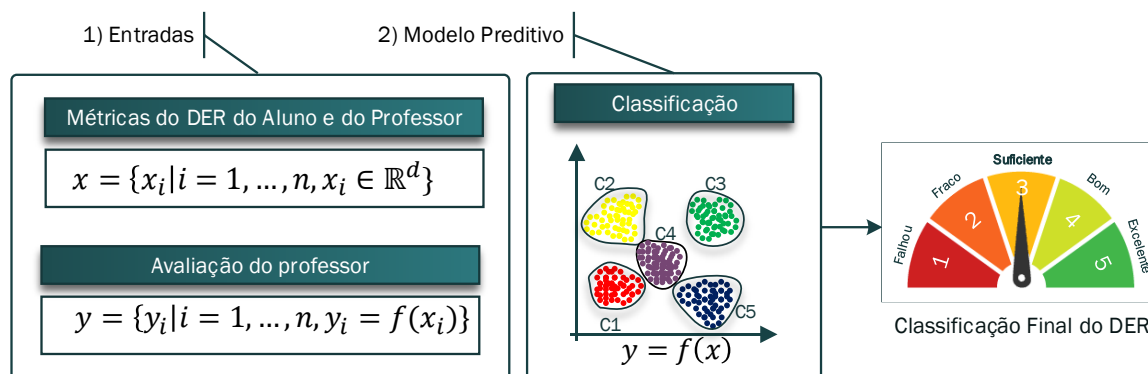


Figura 4-12. Entrada e saída da fase aprendizagem computacional - DER.

Foram analisadas diferentes escalas de avaliação, que variam entre 0 a 10, 0 a 20, entre outros, e optou-se pela escala de 5 pontos (ver Tabela 4-4) por ser uma escala genérica e conhecida no meio académico pelos seus rótulos linguísticos, e ainda por ter uma correspondência direta noutras línguas. A seleção de rótulos de números ímpares tem sido sugerida na literatura, pois contribuem para obter melhores resultados pelo facto de serem equilibrados (Bonissone, 1980; Godo, Mántaras, Sierra, & Verdaguer, 1989).

Tabela 4-4. Rótulos linguísticos de compreensão¹.

Categoria		Valor
Falhou	(0% - 20%)	1
Fraco	(21% - 49%)	2
Suficiente	(50% - 69%)	3
Bom	(70% - 89%)	4
Excelente	(90% - 100%)	5

Então, a partir da aplicação do processo CRISP-DM, o modelo preditivo é concebido, e é capaz de prever a avaliação do professor em cada resposta submetida pelos alunos. A classificação então é tida pelo *framework* como um *feedback* final do sistema. Este *feedback* ajuda os alunos a tomar rapidamente a decisão no que diz respeito a continuar a tentar melhorar a sua resposta ou não. Normalmente o aluno quer sempre obter a melhor classificação possível de entre todas as existentes, pelo que o *feedback* o ajuda nessa tarefa de decisão.

Por outro lado, para o professor, esse *feedback* serve como indicador final do desempenho dos alunos, sendo utilizado para processar por computador a nota final do aluno dentre os diversos exercícios que compõe uma avaliação, bem como servir de evidência de desempenho dos alunos. Lembramos que o sistema permite sempre ao professor atualizar a nota do aluno, e em caso de desacordo com a avaliação automática, pode atribuir outra classificação.

O resultado da classificação tem muitas utilidades para o sistema, por exemplo, pode fornecer um relatório analítico com as classificações dos alunos para ajudar o professor a tomar a decisão de repetir um tópico de aula e até adiar a prova perante um cenário de baixo rendimento. Para os alunos, o relatório ajudaria a focar os estudos no tópico onde têm mais dificuldades.

¹ Para analisar os dados coletados no ensaio, atribuímos um número entre 1 e 5 a cada rótulo linguístico. O pior caso leva o valor um (falhou), e o melhor caso tem valor um (Excelente).

Ambiente de Ensino e Aprendizagem LabDER

Este capítulo descreve o processo de desenvolvimento e teste do ambiente LabDER, apresenta seus requisitos e como eles estão alinhados com os objetivos principais do ambiente proposto, bem como, o processo sistemático de teste para validação e verificação do sistema.

5.1 Processo de Desenvolvimento do LabDER

O LabDER é uma nova versão do ambiente virtual de ensino e aprendizagem de SQL, que originalmente foi concebido para avaliar automaticamente as questões SQL e servir como uma ferramenta de apoio aos professores e alunos no processo de ensino e aprendizagem nas disciplinas de bases de dados de cursos de licenciatura e pós-graduação em informática (A. D. P. Lino, 2007). Assim, o LabDER herda um conjunto de funcionalidades relacionadas com o ensino de bases de dados e adiciona uma série de novas funcionalidades que permitirão que o ambiente cumpra o seu objetivo para esta investigação.

O principal objetivo do LabDER é disponibilizar um ambiente online de aprendizagem de DER que forneça a avaliação automática do DER. Com este propósito, o desenvolvimento resultou em duas versões:

- 1) A primeira versão do ambiente LabDER tem como objetivo a recolha dos dados das respostas dos alunos no contexto de sala de aula, que incluem questões de DER, SQL, escolha múltipla e de texto livre. Os dados de DER devem estar adequadamente armazenados na BD para fornecer o desenvolvimento do modelo de avaliação automática do DER e, posteriormente, serem adicionados ao ambiente.
- 2) A segunda versão do LabDER tem o objetivo de avaliar o DER automaticamente e fornecer o *feedback* semântico com classificação final.

Este capítulo descreve o processo de desenvolvimento da primeira versão, enquanto a segunda versão está descrita nos resultados dos ensaios no capítulo 6.

Portanto, nesta versão, o LabDER é destinado a ser um ambiente de ensino e aprendizagem de BD, que contemple também a avaliação automática de DER, sendo que a avaliação automática fica disponível após o desenvolvimento do modelo preditivo. Assim, os alunos serão submetidos a realizar um conjunto de atividades, incluindo exercícios e exames previamente definidos pelo professor com base no seu plano de ensino. O plano de ensino do professor é a chave para determinar os momentos de aprendizagem e as diferentes modalidades de interação com o ambiente. O ambiente pode ser utilizado à distância ou presencialmente nas aulas teóricas ou práticas como ferramenta para acompanhamento da aprendizagem dos alunos e como recurso para controle do cumprimento do plano de aula do professor.

Para garantir o cumprimento e controle dos requisitos, o desenvolvimento do LabDER seguirá um processo simples de desenvolvimento de software. Nesta tese, referimo-nos a um processo de desenvolvimento de software como uma série de ações, procedimentos ou atividades realizadas pelo investigador e *stakeholders* (professores e alunos) para projetar e produzir um produto de software. O processo envolve traduzir as necessidades do utilizador em requisitos de software, transformando os requisitos de software num projeto, codificando o projeto, testando o código, instalando e verificando o software para uso operacional (IEEE Computer Society, 2017). Em termos mais simples, pode definir-se como um processo pelo qual as necessidades do utilizador são traduzidas num produto de software.

Existem na literatura centenas de modelos e processos de desenvolvimento de software. No entanto, apenas alguns deles se tornaram amplamente aceites hoje em dia. Entre eles estão o modelo cascata (Royce, 1987), V-Model (Park & Choi, 2012), modelo espiral (Boehm, 1986), modelo Iterativo (Stephens Rod, 2015), processo unificado (Kruchten, 2004), arquitetura dirigida pelo modelo (Mellor, Scott, Uhl, & Weise, 2002), desenvolvimento ágil (Weilkiens Tim, Jesko G. Lamm, Stephan Roth, & Markus Walker, 2015), desenvolvimento orientado ao teste (Beck, 2003), Crystal (Cockburn, 2005), Scrum (Schwaber & Sutherland, 2012), XP (Blom, 2010; Tracz, 2002), Lean (Oppenheim, 2011) e Kanban (Ahmad, Markkula, & Oivo, 2013; Kniberg & Skarin, 2010).

O modelo cascata é estruturado como uma cascata de uma fase para outra. O modelo define as fases de desenvolvimento de software, através das quais se passa de forma sequencial. Somente quando a fase anterior estiver totalmente concluída, se prossegue para a próxima. Entretanto, na prática, nunca é um modelo linear simples. O modelo foi aprimorado ao fornecer

uma interação de *feedback* para que cada estágio anterior pudesse ser revisto e sugere-se que uma execução protótipo seja feita antes do projeto principal (Sommerville, 2016).

As fases do modelo cascata refletem diretamente as atividades fundamentais de desenvolvimento de software:

1. Análise e definição de requisitos: As funcionalidades, restrições e metas do sistema são estabelecidos por consulta aos utilizadores do sistema. Eles são definidos em detalhe e servem como uma especificação do sistema.
2. Projeto de sistema e software: Nesta fase estabelece-se uma arquitetura geral do sistema, onde se associa os requisitos aos sistemas de hardware e software. O projeto de software envolve identificar e descrever as abstrações fundamentais do sistema de software e os seus relacionamentos.
3. Codificação: Durante esta fase, o projeto do software é realizado como um conjunto de programas ou unidades de programa.
4. Teste do sistema: As unidades ou programas são integrados e testados como um sistema completo para garantir que os requisitos de software tenham sido tidos em conta.
5. Implantação e Operação: O sistema é instalado e colocado em uso prático. A manutenção envolve a correção de erros que não foram descobertos em estágios anteriores do ciclo de vida, melhorando a implementação de unidades do sistema e aprimorando os serviços do sistema.

A Figura 5-1 ilustra a adaptação do modelo cascata de Sommerville (2016) que associa a fase de teste a um subprocesso para execução do teste de software.

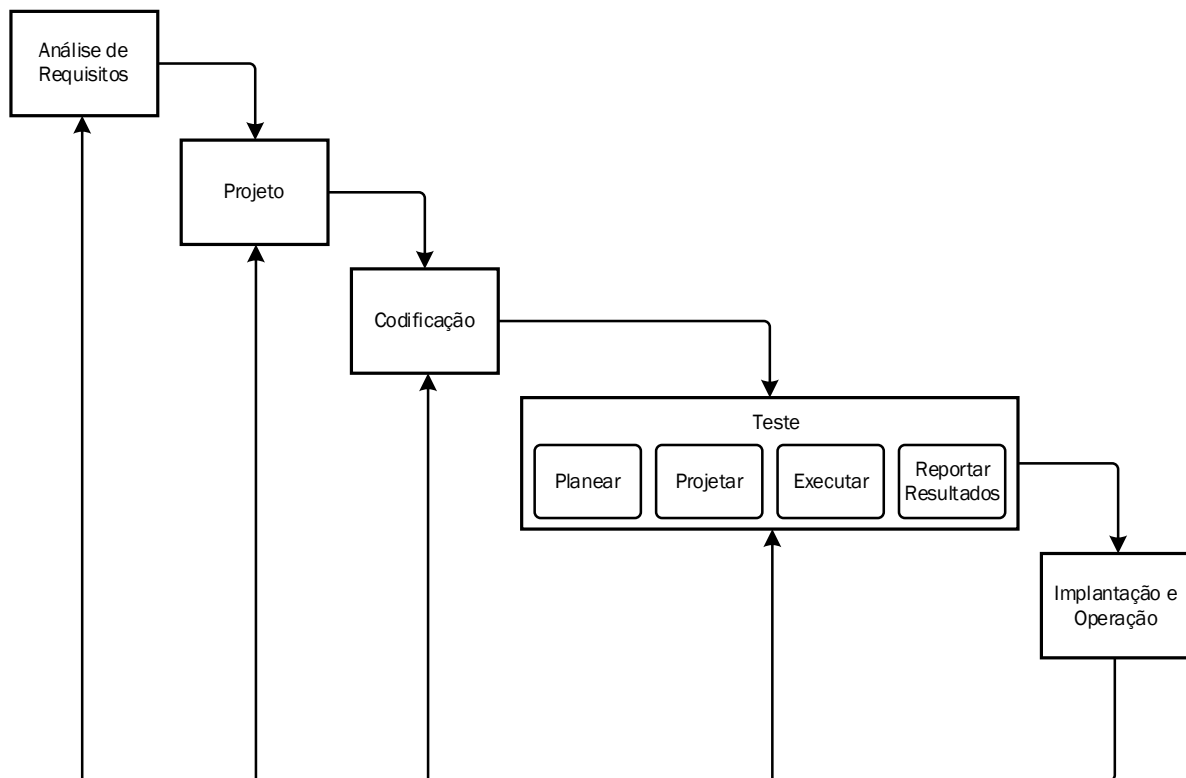


Figura 5-1. Modelo de desenvolvimento cascata, adaptado de (Sommerville, 2016).

Este modelo foi escolhido porque é recomendado para projetos de engenharia de software, pois os requisitos não mudam durante o desenvolvimento do sistema. Mesmo que a fase de teste seja a penúltima, isso não implica perdas, considerando que, neste ponto, as falhas de projeto são as que melhoram a qualidade do sistema. O projeto LabDER alinha-se a este contexto, pois já possui um propósito bem definido de acordo com os objetivos que pretende alcançar e tem baixa probabilidade de incluir alterações no decorrer do projeto.

5.1.1 Análise de Requisitos

A análise de requisitos é o primeiro passo no processo de desenvolvimento de software e engloba as tarefas que determinam as necessidades ou condições a ter em conta num produto novo ou alterado, considerando os requisitos das partes interessadas (Strickland, 2011). Os requisitos podem ser funcionais e não funcionais e devem ser acionáveis, mensuráveis, testáveis, relacionados com necessidades ou oportunidades e definidos com um nível de detalhe suficiente para o projeto do sistema (Strickland, 2011).

A lista de requisitos funcionais e não funcionais do LabDER é extraída da definição do intuito do sistema. A definição do intuito, dá a todos os envolvidos um entendimento comum do resultado do produto, bem como o limite entre o que está dentro e o que está fora de uma determinada versão (Wieggers & Beatty, 2013).

5.1.1.1 Definição do escopo

O produto de software resultante deste processo de desenvolvimento deveria ser um sistema web destinado a apoiar o ensino e aprendizagem sobre bases de dados relacionais. Os utilizadores são alunos e professores, que têm acesso ao sistema após aceitar a política de privacidade e os termos de condições do site que seguem o Regulamento Geral sobre a Proteção de Dados Pessoais (RGPD), conforme o Regulamento n° 2016/679, de 27 de abril de 2016 (Parlamento Europeu, 2016). O sistema herda um conjunto de funcionalidades da versão anterior como por exemplo: gerir utilizadores, gerir questões de texto livre, escolha múltipla e SQL, disponibilizar a avaliação automática do SQL, gerir avaliações, relatórios de desempenho e de acesso, gerir conteúdo, e apresentar área de execução de atividades e visualização de conteúdo.

Nesta versão, o sistema deve disponibilizar meios para desenhar o DER na notação *crow's foot* com todos os recursos para inserir, editar e excluir objetos da estrutura do diagrama ER, bem como disponibilizar todas as transações necessárias para o acompanhamento dos alunos. O professor pode consultar relatórios que são gerados com base nas respostas desenvolvidas e no desempenho alcançado pelos alunos. O professor deve avaliar cada questão e atribuir uma classificação, bem como pode inserir um *feedback*, sugerindo melhorias, quando apropriado.

Como premissa para utilização, é esperado que o professor alimente o sistema com o plano de ensino, e importe a lista de alunos utilizando arquivos no formato CSV. Essa prática simplifica a alimentação com os dados dos alunos. Entretanto, ao professor também é dada a possibilidade de incluir manualmente novos alunos através de formulários, se assim preferir. Em geral o professor deve ter permissão para gerir os alunos, turmas, cursos, disciplina, questões e avaliações, bem como poderá alterar o material didático padrão do sistema, que inclui um livro no formato digital com 6 capítulos.

Também é papel do professor definir as estratégias para criar as avaliações: Se serão compostas por questões aleatoriamente selecionadas pelo sistema ou conterão o mesmo conjunto de questões para todos os alunos. O recurso para criar avaliações aleatórias é normalmente utilizado em dias de provas para evitar burla, uma vez que os laboratórios de informática possuem computadores muito próximos uns aos outros. Por outro lado, as avaliações com questões iguais a todos os alunos são normalmente aplicadas nas listas de exercícios. O sistema ainda deveria usar o conceito de grupo de questões para criar uma avaliação com questões aleatórias ou idênticas aos alunos. O professor poderá definir

atividades em equipa e o sistema automaticamente replicar as questões para todos os membros da equipa.

O aluno, por sua vez usa as suas credenciais de login e senha para aceder ao sistema através da interface, e conclui a sua inscrição confirmando a sua participação na turma. Ao confirmar a sua inscrição é criada uma base de dados para salvar os desenhos de DER e testar comandos SQL. Então, o sistema cria a base de dados individual do aluno e armazena o seu consentimento e a sua autorização de recolha de dados durante a visualização da política de privacidade e os termos de uso do sistema.

O sistema deve apresentar as avaliações disponíveis no período de acesso, no caso de existir uma avaliação em curso, o sistema deve bloquear o acesso às demais funções do sistema como, por exemplo, aceder ao material didático. Mas, caso não existam avaliações disponíveis, o sistema deve apresentar o menu principal e fornecer acessos aos demais itens, como material de apoio e acesso aos relatórios.

O aluno pode fazer exercícios e avaliações a fim de solucionar as questões propostas pelo professor do tipo de texto livre, escolha múltipla, SQL e DER, sendo as respostas armazenadas na BD principal do sistema, onde estão armazenados os dados de desempenho de todos os alunos. A interface de exercícios e avaliações deve ser dinâmica, ou seja, adaptar-se automaticamente à maneira como os utilizadores interagem com os diferentes tipos de questões. Ainda o aluno dispõe do módulo de visualizar material didático, que semelhante a um livro, fornece um conjunto de exemplos para os alunos testarem.

O sistema deve ter funções de modo treino para DER e SQL que não estejam obrigatoriamente vinculadas aos exercícios ou exames propostos pelo professor. Além disso, o sistema deve registar todas as interações dos utilizadores no sistema em forma de log, e permitir rastrear a navegabilidade no sistema, bem como poder identificar o nível de software e hardware utilizado pelo utilizador para aceder ao sistema e à localização do acesso.

O sistema deve permitir o acesso dos utilizadores de qualquer localidade e deve garantir a integridade e segurança dos dados para impedir registo duplicados e problemas com login. Todas as transações do ambiente devem atender o tempo de resposta que a literatura apregoa em termos de performance de sistemas web. Também é necessário que o sistema exporte todos os DER no formato SQL para posterior processamento.

5.1.1.2 Definição de requisitos

A partir da descrição do propósito é gerada uma lista de requisitos. Um requisito é uma declaração sobre o que o sistema deve ser capaz de fazer. Este tipo de requisito é chamado de requisito funcional, isto é, é a descrição de uma funcionalidade que o sistema deve assistir. Há também o requisito que espelha algo sobre o próprio sistema ou quão bem ele executa as suas funções. Tais requisitos são frequentemente chamados de requisitos não funcionais, ou requisitos de qualidade e incluem atendimento de características quanto a usabilidade, disponibilidade, confiabilidade, capacidade de suporte, testabilidade, facilidade de manutenção e facilidade de uso (Strickland, 2011).

Como mencionado na seção 5.1, o LabDER contempla um conjunto de funcionalidades herdadas do LabSQL, bem como um conjunto de funcionalidades identificadas a partir da necessidade de disponibilizar um ambiente de ensino e aprendizagem de bases de dados relacionais, que inclua a avaliação automática de DER.

A análise dos requisitos para o desenvolvimento do LabDER resultou numa lista de requisitos funcionais (RF) e requisitos não funcionais (RNF), totalizando 38 RF, sendo 25 RF do perfil do professor (ver Tabela 5-1), 13 RF do perfil aluno (ver Tabela 5-2), e 5 RNF do sistema (ver Tabela 5-3).

Tabela 5-1. Lista de requisitos funcionais do perfil do professor.

Perfil	Id. RF	Requisito Funcional	Descrição
Professor	RF 01	Login	O sistema deve permitir o acesso ao sistema de professores autorizados.
	RF 02	Logout	O sistema deve permitir a saída segura do sistema.
	RF 03	Gerir Alunos	O sistema deve permitir registar, editar, visualizar e importar informações referentes ao aluno.
	RF 04	Gerir Equipas	O sistema deve permitir registar, editar e visualizar informações referentes aos grupos de alunos.
	RF 05	Gerir Avaliações	O sistema deve permitir registar e editar avaliações.
	RF 06	Libertar Avaliações	O sistema deve permitir ativar e desativar as avaliações por aluno, e também permitir edição das avaliações ativas num intervalo de tempo especificado.
	RF 07	Visualizar relatórios	O sistema deve permitir visualizar relatórios de acesso dos alunos por mês e acessos da turma sumariado, e também das avaliações dos alunos em diferentes formatos com a possibilidade de alteração de nota.
	RF 08	Gerir Questões	O sistema deve permitir registar e editar questões: de texto livre, de escolha múltipla, SQL e DER. Também deve permitir exportar questões no formato PDF.
	RF 09	Gerir Grupos Questões	O sistema deve permitir registar e editar questões, grupos de questões.
	RF 10	Gerir Respostas	O sistema deve permitir registar e editar a resposta do professor das questões de texto livre, de escolha múltipla, SQL e DER.
	RF 11	Gerir Professor	O sistema deve permitir registar, editar e visualizar professor.
	RF 12	Gerir Instituição	O sistema deve permitir registar, editar e visualizar instituição de ensino.
	RF 13	Gerir Tipo de Curso	O sistema deve permitir registar, editar e visualizar tipo de curso.
	RF 14	Gerir Curso	O sistema deve permitir registar, editar e visualizar curso.
	RF 15	Gerir Disciplina	O sistema deve permitir registar, editar e visualizar disciplina.
	RF 16	Gerir Turma	O sistema deve permitir registar, editar, visualizar e trocar de turma.
	RF 17	Gerir Conteúdo	Permite ao professor alterar o conteúdo do tópico de bases de dados.
	RF 18	Material de Apoio	Permite ao professor visualizar uma lista de material de apoio para download.
	RF 19	SQL - Livre	O sistema deve permitir a execução de scripts SQL numa instância de BD MySQL.
	RF 20	DER - Livre	O sistema deve permitir criar, editar, salvar e exportar diagramas ER.
	RF 21	Verificar BD	O sistema deve permitir restaurar a imagem dos BD dos alunos.
	RF 22	Registar nota	O sistema deve permitir ao professor atribuir uma nota na resposta do aluno.
	RF 23	Feedback automático	O sistema deve permitir ao professor consultar o <i>feedback</i> da resposta submetida pelo aluno.
	RF 24	Feedback manual	O sistema deve permitir ao professor criar um <i>feedback</i> manual na resposta de DER submetida pelo aluno.
	RF 25	Termos de uso	O sistema deve permitir o acesso após concordância com os termos da política de privacidade seguindo o RGPD e consentir o uso de dados.

Tabela 5-2. Lista de requisitos funcionais do perfil do aluno.

Perfil	Id. RF	Requisito Funcional	Descrição
Aluno	RF 26	Login	O sistema deve permitir o acesso ao sistema de alunos autorizados.
	RF 27	Logout	O sistema deve permitir a saída segura do sistema.
	RF 28	Gerir Aluno	O sistema deve permitir visualizar e editar dados dos alunos
	RF 29	Verificar BD	O sistema deve permitir ao aluno verificar e restaurar a sua BD.
	RF 30	Visualizar relatórios	O sistema deve permitir aos alunos visualizar relatórios de frequência do acesso e relatório das avaliações realizadas.
	RF 31	Visualizar Avaliações	O sistema deve permitir ao aluno visualizar a lista de avaliações ativas dentro do período especificado por data e hora.
	RF 32	Exportar Conteúdo	O sistema deve permitir ao aluno exportar as questões e todo o material didático.
	RF 33	Ver Conteúdo	O sistema deve permitir ao aluno visualizar o conteúdo disponível no material didático.
	RF 34	SQL - Livre	O sistema deve permitir a execução de scripts SQL numa instância de BD MySQL e salvar os comandos executados. Também deve mostrar comandos SQL associados aos exemplos didáticos quando o aluno estiver a visualizar o material didático.
	RF 35	DER - Livre	O sistema deve permitir criar, editar, salvar os DER na instância da BD do utilizador, e também exportar o DER no formato SQL.
	RF 36	Responder Questões	O sistema deve permitir ao aluno responder as questões de texto livre, de escolha múltipla, SQL e DER.
	RF 37	Feedback Automático	O sistema deve enviar um <i>feedback</i> da avaliação automática das questões SQL, escolha múltipla e de texto livre.
	RF 38	Termos de uso	O sistema deve permitir o acesso após o concordar com os termos da política de privacidade seguindo o RGPD e consentir o uso de dados.

Os requisitos não funcionais do LabDER são igualmente importantes pois reúnem características de qualidade que vão tornar o sistema mais rápido, fiável e fácil de usar. O não cumprimento desses requisitos, pode levar à desmotivação do utilizador e ao consequente desuso do sistema. Os requisitos não funcionais (RNF) estão listados na Tabela 5-3.

Tabela 5-3. Descrição dos requisitos não funcionais (RNF).

Id. RNF	Descrição
RNF 1	O sistema deve ser web e funcionar nos navegadores Chrome, Mozilla Firefox e Microsoft Edge.
RNF 2	O sistema deve ser capaz de se integrar numa API de desenho de DER.
RNF 3	O sistema deve ter tempo de resposta máximo entre 2 e 10s.
RNF 4	O sistema deve suportar uma carga mínima de 500 utilizadores em simultâneo.
RNF 5	O ambiente deve disponibilizar uma interface de fácil utilização para o utilizador.

5.1.2 Projeto

Esta secção descreve a arquitetura geral do sistema, as configurações necessárias a nível de hardware e software a fim de alcançar os requisitos, bem como, a definição do protótipo do ambiente.

5.1.2.1 Arquitetura

O LabDER utiliza a arquitetura cliente-servidor em três camadas. Na camada de apresentação ou interface estão os professores e alunos como utilizadores, que acedem à aplicação através da internet utilizando um navegador disponível nos seus computadores. Na camada lógica temos um servidor de aplicação com o sistema operacional Ubuntu com o serviço web Apache HTTP para disponibilizar o sistema, desenvolvido na linguagem PHP na versão 5.6.35. Na camada de dados, temos o servidor de bases de dados com o SGBD MySQL na versão 5.7.21 onde estão contidos os dados. Ambos os servidores são virtuais, e estão instalados na Cloud do Departamento de Engenharia Informática da Universidade de Coimbra. A configuração inicial do servidor de aplicação possui 4 unidades virtuais de processador (vCPU) e 4 gigabytes (GB) de memória RAM e um disco de 50 GB, e o servidor de BD possui a configuração de 4vCPU e 4GB de RAM. A Figura 5-2 ilustra a arquitetura geral do sistema.

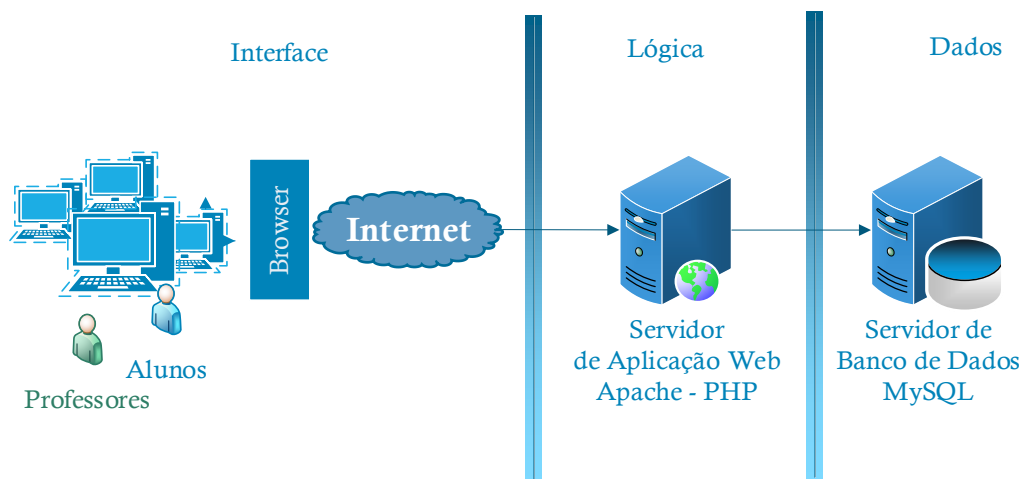


Figura 5-2. Arquitetura cliente-servidor inicial do ambiente LabDER.

A arquitetura de software do LabDER (ver Figura 5-3) ilustra como o projeto foi estruturado em oito módulos para suportar os RF e RNF do sistema, e como esses módulos se relacionam entre si e com os outros componentes. A camada de apresentação representa o navegador que utiliza o protocolo SSL. A camada de aplicação está composta por módulos, componentes e tipos de questões. Os módulos de gestão podem ser vistos como funções gerais para inserir, editar, visualizar e excluir os dados que devem ser contidos na camada de dados. O módulo de exercícios e avaliação comporta os tipos de questões DER, SQL, escolha múltipla e questões de texto livre, que acionam o componente de avaliação automática correspondente. O componente de avaliação automática é acionado para avaliar cada resposta do aluno. No caso de respostas de DER e SQL, a resposta do aluno é comparada à resposta do professor em três níveis: compilador, métricas e o modelo preditivo conforme descrito no capítulo do

framework de avaliação automática de DER e SQL, e usa a BD de métricas para processar todo o volume de dados necessários e servir de suporte para calcular a nota final. A camada de dados está dividida em três grupos de instâncias, a primeira dedicada a conter todos os dados de entrada do sistema na BD do LabDER, a segunda dedicada a conter os dados da avaliação automática na BD de Métricas, e um grupo de instância dedicado às BD dos utilizadores, sendo que cada aluno e professor possuem uma instância de BD independente.

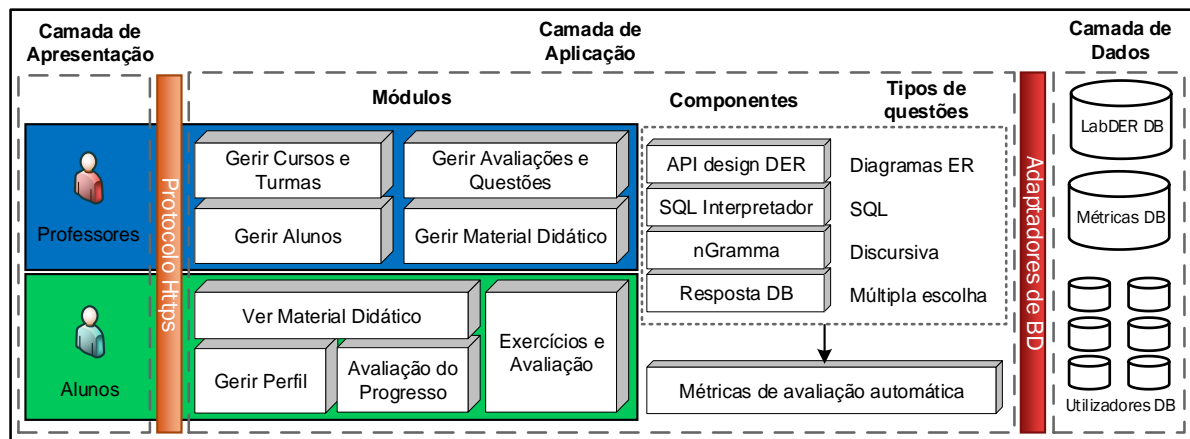


Figura 5-3. Visão geral da arquitetura de software do LabDER.

As diferentes instâncias servem para isolar os acessos em nível de componente e utilizador, que são controlados por prerrogativas específicas de segurança do SGBD. Então, mesmo que um utilizador tenha acesso a BD dos utilizadores, esse utilizador não consegue aceder a outra instância porque não possui as prerrogativas de acesso.

Para disponibilizar uma interface aos alunos para desenhar os DER livremente durante o processo de ensino-aprendizagem, foi necessária a identificação de uma API que tivesse em conta alguns critérios de aceitabilidade, como: a API deve aceitar o desenho online de DER em notação *crow's foot*, a API deve fornecer meios para exportar o DER em formato SQL e conectar a uma base de dados relacional. Outro critério para selecionar a API é o uso de uma tecnologia consolidada, ou seja, características da comunidade são tidas em consideração na análise, como por exemplo o *feedback* dos desenvolvedores em fóruns que utilizam a API e o seu tempo de existência. O último critério é o uso de tecnologias gratuitas e *opensource* para evitar custos adicionais ao projeto, uma vez que essas tecnologias podem ser incorporadas e alteradas sem custos adicionais.

Existiam diferentes APIs disponíveis para desenhar os diagramas online que podiam ser utilizadas e/ou adaptadas para o desenvolvimento do nosso protótipo. Portanto, uma metodologia de investigação foi definida para sistematizar o processo de escolha através dos critérios acima descritos. As palavras-chave utilizadas para encontrar as APIs são: API,

ferramenta, web, online, DER, diagrama, entidade e relacionamento, sistemas, edição de diagramas, desenhos de diagrama, e suas combinações. Diferente das investigações científicas, que normalmente são realizadas em arquivos de dados científicos, essas investigações focam-se em encontrar e testar códigos fontes. Nesse caso, a investigação é realizada em arquivos públicos de código fonte como o GitHub, code.google.com, sourceforge.net e archive.codeplex.com.

O teste das APIs compreende a instalação dos componentes e a sua execução num servidor local, recorrendo aos navegadores Chrome, Mozilla Firefox e Internet Explorer, para verificar se as funcionalidades estão em conformidade. Ainda, faz parte do teste da API identificar o esforço necessário para usar a API para criar, editar e salvar o DER integrado na linguagem PHP. No final desta fase, é avaliado o grau de dificuldade de implementação e extensão da API a um sistema educacional. Neste caso, foi definida a escala difícil, normal e fácil para descrever o grau de dificuldade de implementação.

O período compreendido de investigação das ferramentas foi entre 01 de outubro a 30 de novembro de 2017, e os períodos de testes das APIs foi entre 01 de dezembro de 2017 a 31 de março de 2018.

Na secção de análise de API, são apresentadas alternativas de componentes para que seja desenvolvido o DER com a melhor aparência e facilidade na utilização da biblioteca para desenvolver o protótipo.

5.1.2.2 APIs para desenho de ER

De entre as APIs encontradas durante o processo de investigação, a maioria foi descartada por apresentar problemas durante a instalação de componentes essenciais para a execução da API. Também, uma quantidade significativa de APIs foi descartada por apresentar erros de execução, como por exemplo a funcionalidade de arrastar e soltar não funcionava adequadamente, ou a seleção do objeto não era precisa, e não foi possível corrigir os erros por falta de informações disponibilizadas pela comunidade que dava suporte a essas API. Decidiu-se, portanto, ampliar a procura por ferramentas que pudessem ser integradas ao LabDER.

Os resultados da análise das APIs e ferramentas estão listados na Tabela 5-4. Essa tabela descreve o nome da API, o código fonte utilizado na construção do editor de diagrama, os conceitos de DER envolvidos no desenho, o tipo de notação utilizada no DER, o formato de exportação disponível do desenho, o tipo de distribuição do software, a existência de

funcionalidade para salvar o DER na BD, e por último, uma classificação do grau de dificuldade de implementação da API.

Tabela 5-4. Comparação das APIs para edição de DER.

Nome da API	Código Fonte	Entidade, Atributo e Relacionamento	Notação	Exporta	Distribuição	Salva em BD	Complexidade
MxGraph (2017)	PHP	Sim	<i>Crow's foot</i> e Chen	JPG, XML	OpenSource	Não	Difícil
JGraph - JavaScript Diagramming (2017)	PHP, Javascript	Sim	<i>Crow's foot</i> e Chen	JPG, XML	OpenSource	Não	Média
Creately - Online Diagram Software (2017)	Adobe's Flex/Flash	Sim	<i>Crow's foot</i> e Chen	XML	Proprietário	Não	Difícil
Lucidchart - Online Diagram Software & Visual Solution Lucidchart (2017)	HTML5, JavaScript	Sim	<i>Crow's foot</i> , Chen, Barker's, ISO	PDF, JPG	Proprietário	Não	Difícil
Smartdraw - Create Flowcharts, Floor Plans, and Other Diagrams on Any Device (2017)	HTML5, JavaScript	Sim	<i>Crow's foot</i> e Chen	JPG	Proprietário	Não	Difícil
ERDPlus (2017)	HTML 5, Javascript	Sim	<i>Crow's foot</i> e Chen	XML, SQL	Proprietário	Sim	Difícil
Draw.io (2017)	Html 5, Japascript	Sim	<i>Crow's foot</i> e Chen	XML	Proprietário	Não	Difícil
Laravel Database Projetoer (Prakosa Budi, 2015)	Html 5, PHP	Sim	Bachman	PHP	OpenSource	Não	Média
MindFusion - JavaScript Diagram Library Drawing Library For Flowcharts (2017)	Javascript	Sim	<i>Crow's foot</i> , Chen, Bachman	JPG	Proprietário	Sim	Difícil
SqlDbm - SQL Server Database Modeler, Entity Relationship Diagram (2017)	Javascript	Sim	IDEFIX	SQL	Proprietário	Sim	Fácil
QUICKDBD - Draw database diagrams by typing (2017)	HTML5	Sim	Barker's	SQL, PNG, PDF	Proprietário	Sim	Fácil
Dbprojetoer (Zhao, Sviridenko, Torgersen, Vliet, & Nianios, 2017)	HTML5	Sem Cardinalidade	Bachman	SQL	Proprietário	Sim	Fácil
Wwwsqlprojetoer (Ondřej Žára, 2012)	Javascript	Sim	<i>Crow's foot</i>	SQL	OpenSource	Sim	Fácil

A API considerada mais robusta em relação às demais é a “wwwsqldesigner”. Os motivos são, o facto de possuir um código fonte que não é tão complexo de manter, as funcionalidades de salvar o diagrama na BD e exportá-lo na linguagem SQL. Essas funções estão disponíveis e adaptam-se com facilidade a qualquer outra aplicação, outra vantagem é o uso da notação *crow's foot*, visto que é aceite pela comunidade académica na modelação física e largamente utilizada nas ferramentas CASE. Por fim, devido ao apoio dado a este projeto pela comunidade openSource, há mais de 10 anos, providenciando o suporte de manutenção dessa API.

5.1.2.3 Protótipo

O protótipo do LabDER foi projetado em conformidade com os requisitos para permitir o desenvolvimento de diferentes tipos de questões, em especial questões de DER, onde o aluno

puddesse ter um ambiente livre e fácil para desenho, bem como disponibilizar ao professor, um meio de avaliar e fornecer seu *feedback*. Esta secção define o protótipo do editor que consiste em quatro painéis (ver Figura 5-4): (1) painel de questões, (2) painel de desenho, (3) painel de *feedback*, e (4) painel de histórico.

The screenshot shows the LabDER interface in a Mozilla Firefox browser. At the top, there is a navigation bar with question numbers: 278, 283, 284, 307, 493, 495, 500, 516, 535, 546, 548, 553. Below this, a user profile and session information are displayed: '3ª Frequência Início:11/06/2019 11:00:00 - Término:11/06/2019 12:30:00'. The main area is divided into four panels:

- Panel 1 (Top):** Contains the question text: '284-Crie um diagrama relacional para armazenar informação de uma galeria de artistas. Observe, em particular, as seguintes regras de negócio que são refletidas nas tabelas e seus campos: a) Um pintor realiza uma ou muitas pinturas. b) Cada pintura é pintada por apenas um pintor. c) Uma galeria exibe, obrigatoriamente, uma ou muitas pinturas. d) O pintor só pode exibir a sua pintura numa única galeria a título permanente.'
- Panel 2 (Middle):** Displays an Entity-Relationship Diagram (ERD) with three tables: PINTOR, GALERIA, and PINTURA. PINTOR has fields PINTOR_NUMERO (PK), PINTOR_LNAME, PINTOR_FNAME, and PINTOR_INITIAL. GALERIA has GALERIA_NUM (PK), GALERIA_NOME, and GALERIA_WEB. PINTURA has PINTURA_NUM (PK), PINTURA_TITULO, PINTOR_NUMERO (FK), and GALERIA_NUM (FK). Relationships are shown with lines and crow's foot notation.
- Panel 3 (Bottom Left):** Shows a feedback message: 'O professor atribuiu a avaliação: Nota=4 (70% - 89%) - Bom'.
- Panel 4 (Bottom Right):** Shows a 'Histórico de Respostas' table with columns for question numbers and scores.

	Q.278	Q.283	Q.284	Q.307	Q.493	Q.495	Q.500
Nota	4.00	5.00	4.00	4.00	1.00	1.00	1.00
MP	M.P	M.P	M.P	M.P	M.P	M.P	M.P
1	4	0	0	4.00	5.00	1.00	1.00

Figura 5-4. Editor de diagrama de entidade e relacionamento.

O **painel de questões** (ver item 1 da Figura 5-4) exibe a quantidade de questões disponíveis na avaliação ou no exercício proposto. As questões são identificadas por números, e o aluno pode escolher a questão que pretende solucionar primeiro. O aluno não é obrigado a responder aos exercícios pela ordem em que aparecem as questões, e a sua escolha não influencia a avaliação. Ao clicar na questão, o aluno visualiza a descrição do problema que fornece detalhes do projeto DER. Ainda neste painel, o aluno visualiza as datas de início e de término da atividade proposta que é controlada com informações do servidor. As submissões são processadas apenas dentro do intervalo de data e hora especificado para aquela atividade, quer seja um exercício ou uma avaliação.

O **painel de desenho** (ver item 2 da Figura 5-4) exibe o diagrama DER do projeto do aluno. Nesse painel, o diagrama é desenhado ou atualizado sempre que o aluno interage com o mesmo, clicando e arrastando os componentes livremente sem nenhuma interferência da interface. O aluno pode inserir, editar ou excluir tabelas, atributos e relacionamentos nesse painel. Também pode alterar o tipo de dado dos atributos e definir os campos chave. Para todas essas funções existe uma lista de atalhos que podem ser usados para desenhar o DER mais rapidamente, por exemplo, é possível usar a tecla ‘T’ para adicionar uma tabela, a tecla ‘A’ para adicionar um atributo, a tecla ‘P’ para adicionar uma chave primária, entre outras. Através deste painel, o aluno também pode exportar o diagrama para a linguagem SQL. Os relacionamentos podem ser alterados de quatro maneiras, dando um duplo click na linha e aparece uma janela com todas opções disponíveis, clicando em cada extremidade da linha da relação, clicando na linha e depois na opção editar relacionamento ou usando a tecla de atalho ‘R’. Neste painel, o aluno pode nomear as tabelas e os atributos da forma que bem entender, tal como também pode adicionar anotações ou comentários às tabelas ou ao diagrama.

Os alunos podem salvar o diagrama, através do botão ‘Enviar’, o que salva automaticamente a sua resposta na BD e atualiza primeiramente o painel de *feedback* e em seguida o painel de histórico.

A finalidade do **painel de *feedback*** (ver item 3 da Figura 5-4) é a de disponibilizar um espaço para apresentar aos alunos, as respostas às questões. No caso, das questões de texto livre, escolha múltipla e SQL, a avaliação automática devolve automaticamente um *feedback* para cada questão. Para questões do tipo DER, o painel apresenta a classificação registada pelo professor.

A finalidade do **painel do histórico** (ver item 4 da Figura 5-4) é a de permitir ao aluno visualizar todas as respostas que foram submetidas. O aluno pode visualizar o histórico clicando na sua resposta, que é automaticamente restaurada para a interface e reprocessada. Caso o aluno faça modificações no diagrama e salve posteriormente, essa submissão é processada como uma nova resposta e incrementada no histórico da avaliação ou do exercício.

5.1.3 Desenvolvimento

O LabDER foi desenvolvido na linguagem PHP e hospedado no serviço da Cloud do Programa de Doutoramento em Ciências e Tecnologias da Informação da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC), disponível nas versões Português e Inglês (ver Figura 5-5) pelo link <https://labder.dei.uc.pt/>.



Figura 5-5. Interface de login do LabDER.

A versão do PHP utilizada durante o desenvolvimento do LabDER é incompatível com a versão do LabSQL e, por isso, todas as funções herdadas tiveram que ser reescritas. Não foi encontrada uma forma de automatizar o processo de transição de versão, e por isso o processo de migração foi manual, ou seja, todo o código fonte e qualquer evento do LabSQL teve de ser reescrito um a um.

O LabDER possui mais de 300 MB de código fonte, incluindo as APIs que foram incorporadas. Ao todo, são 28.271 arquivos com cerca de 2.180 diretórios. A versão anterior possuía 7.412 arquivos e 623 diretórios e ocupava 183 MB de espaço em disco.

O desenvolvimento foi feito em módulos e de forma progressiva por perfil de utilizadores e será apresentado dessa forma nas secções seguintes, inicialmente pelo perfil do professor, e em seguida o perfil do aluno.

5.1.3.1 Perfil do professor

No perfil do professor, as funcionalidades do sistema são agrupadas para representar as diferentes fases de execução de um curso de BD. Por exemplo, numa primeira fase, a preparação do curso pode ser vista como o grupo de funcionalidades para criar uma turma, depois inserir os alunos, em seguida registar as questões e as respostas, as respetivas avaliações e, por fim, o acompanhamento do progresso dos alunos pelos relatórios. Estas atividades estão agrupadas em cinco módulos, conforme a ilustração do painel do menu principal do professor (ver Figura 5-6).

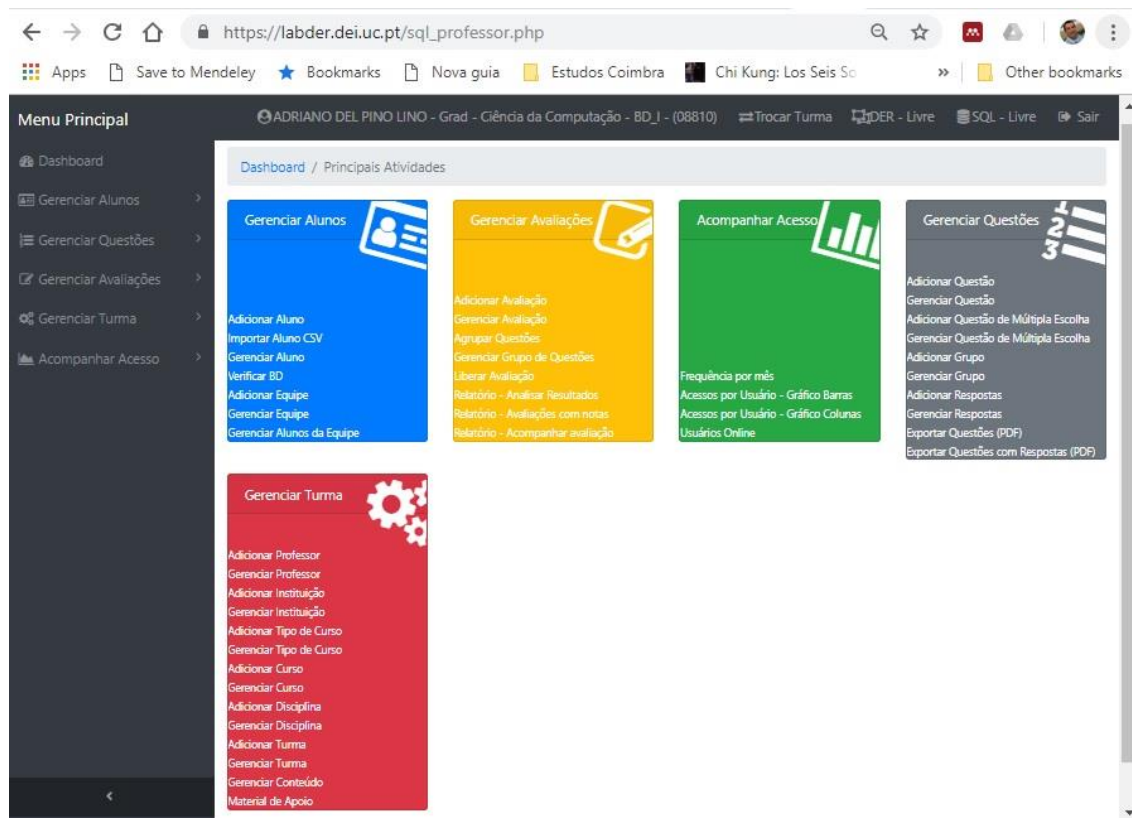


Figura 5-6. Painel principal - Perfil do professor.

Os cinco módulos apresentados no painel principal do professor estão associados aos grupos de funcionalidades que se relacionam entre si (ver Figura 5-6), eles são:

1. Gerir alunos: Este módulo gere as informações dos alunos; os recursos são: inscrever alunos, por importação de arquivo CSV ou registo manual do aluno através de um formulário padrão, gerir as equipas de alunos para executar atividades e gerir a BD do aluno. Os alunos têm acesso ao sistema depois de concordar com os termos legais de uso e a política de privacidade disponível no site do LabDER.
2. Gerir avaliação: Este módulo gere as avaliações e os grupos de questões que fazem parte das avaliações. O recurso de gestão de grupos de questões é usado para providenciar a opção de criar atividades com questões aleatórias. Normalmente, as listas de exercícios possuem as mesmas questões para todos os alunos e devem usar todas as questões do grupo. Os exames são geralmente aleatórios e devem usar uma quantidade menor de questões, em grupo, para serem aleatórios, o que evita o plágio. Ainda é possível ativar ou desativar uma avaliação e visualizar os relatórios para acompanhar e apreciar / qualificar as avaliações dos alunos. Os relatórios permitem analisar o progresso do aluno e avaliar manualmente as suas atividades, e no caso de DER, é possível incluir um *feedback* no formato de post-it virtual.

3. Acompanhar o acesso: Este módulo apresenta os relatórios de acesso dos utilizadores, que incluem informações de frequência de acesso e utilização do sistema.
4. Gerir questões: Este módulo gere as questões e respostas. Existem quatro tipos de questões: de texto livre, escolha múltipla, SQL e DER na notação *crow's foot*. Os professores podem partilhar questões e respostas uns com os outros, e exportá-las em PDF.
5. Gerir turma: Este módulo prepara o LabDER para ser usado pelo professor e por uma turma de alunos. Os recursos de gestão de turma permitem criar, editar, visualizar: professor, instituições de ensino, curso, disciplina e turma. Também inclui a gestão de material didático, onde o professor pode editar o livro digital padrão que vem com o sistema e adicionar outros conteúdos digitais como apresentação.

Outras funções do painel principal do professor são: SQL-Livre e DER-Livre. O SQL-Livre é uma interface para se conectar a uma instância da BD do utilizador utilizando o MySQL e executar códigos SQL. O DER-Livre é uma interface para criar desenhos DER na notação do *crow's foot*, que permite salvar o desenho na BD do utilizador ou exportar para o SQL. Durante as aulas, o professor pode explicar os conceitos de BD recorrendo a estes recursos.

O painel de relatório para analisar resultados (ver Figura 5-7) fornece uma lista de alunos com a nota final obtida numa determinada avaliação previamente seleccionada. Como o sistema permite várias respostas por questão, o relatório apresenta a melhor pontuação obtida por questão. As avaliações podem ter diferentes escalas, e o sistema encarrega-se de processar a nota final e apresentá-la no relatório.

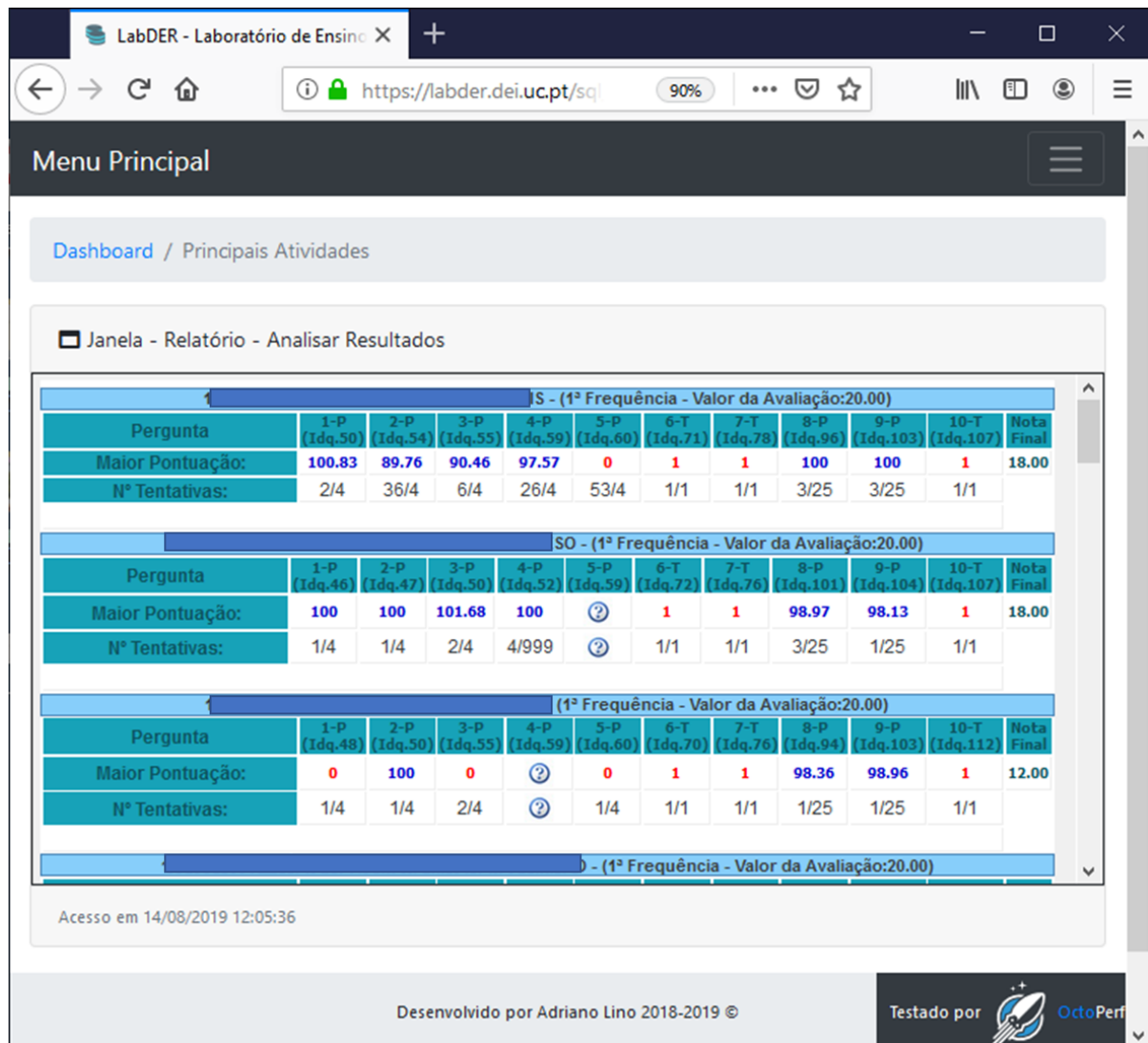


Figura 5-7. Relatório analisar resultados - Perfil do professor.

As notas e os nomes dos alunos apresentados na avaliação do relatório (ver Figura 5-7) fornecem links de acesso para aceder à página do aluno (Figura 5-8). Desta forma, o professor pode visualizar as respostas submetidas e, em caso de desacordo com a avaliação automática, submeter uma avaliação manual. O professor também consegue visualizar (ver item 1 na Figura 5-8) o conjunto de questões da avaliação e navegar entre cada uma das respostas submetidas (ver item 2 na Figura 5-8) e ver o mesmo *feedback* que foi recebido durante a submissão do aluno (ver item 3 na Figura 5-8). Ainda pode navegar nas diversas respostas submetidas pelo aluno no painel de histórico de resposta (ver item 4 na Figura 5-8).

LabDER - Laboratório de Ensino e Aprendizagem de DER - Exercícios - Mozilla Firefox

https://labder.dei.uc.pt/sql_exercicio.php?cdProb=127&cdResposta=1&ida314&cdConteudo... 90%

Questão 127 128 131 161 163 166 171 468 478 482

2ª Frequência Início:16/04/2019 10:00:00 - Término:16/04/2019 11:25:00

166-Mostre a media dos salários de cada depto.

```

1 select depto.nome, AVG(salario)
2 from depto join emp on depto.cddepto=depto
3 group by depto.nome
4

```

Alterar Nota Máx.Pontos:100.00

OBS: Seu SQL está 91,14% correto.
Tente novamente para alcançar a solução ótima.

Dica: Verifique os seguintes elementos de sintaxe para alcançar a solução ótima:
Excedeu sintaxe de operador (on, join) ; Faltou sintaxe de operador (where)

```

select depto.nome, AVG(salario) from depto join emp on
depto.cddepto=depto group by depto.nome

```

nome	AVG(salario)
Computação	24333.3333
Sede	29000.0000
Seguranca	21000.0000

3 row(s)

Histórico de Respostas					
Q.127	Q.128	Q.131	Q.161	Q.163	Q.166
112.19	110.93	99.59	0.00	0.00	0.00
			0.00	0.00	0.00
			0.00	0.00	0.00
			0.00	0.00	0.00
			102.43		101.83
M.P.	M.P.	M.P.	M.P.	M.P.	M.P.
112.19	110.93	99.59	102.43		101.83

Figura 5-8. Janela de execução de exercício SQL - Perfil do professor.

No caso de exercícios de DER, a área de *feedback* pode dar a informação referente a diversas situações do estado da avaliação, se está pendente, se acabou de avaliar ou se já avaliou. O professor pode avaliar o DER recorrendo aos cinco rótulos linguísticos, sendo que cada rótulo possui limites mínimo e máximo de representatividade (Falhou 0% a 20%, Fraco 21% a 49%, Suficiente 50% a 69%, Bom 70% a 89% e Excelente 90% a 100%), como pode ser visto na Figura 5-9.

Uma vez que o professor atribui uma classificação na resposta do diagrama ER do aluno, o sistema emite uma mensagem “alterou a avaliação para:”, juntamente com a nova escala. O professor pode ainda adicionar um *feedback* descritivo em formato de *post-it* virtual, sendo possível justificar a classificação atribuída ou indicar as correções necessárias para dar ao aluno a possibilidade de realização de novas correções. Na Figura 5-9 o *feedback* descritivo contém a mensagem “Faltou adicionar o tipo de cardinalidade”, referindo-se à ausência de cardinalidades no DER submetido pelo aluno.

Feedback manual: O professor pode enviar uma explicação em forma de ticket

268-Crie o diagrama relacional para o banco de dados mostrado na Figura Q3.8.

STUDENT
STU_CODE INTEGER PK
PROF_CODE INTEGER FK

PROFESSOR
PROF_CODE INTEGER PK
DEPT_CODE INTEGER PK

Faltou adicionar o tipo de cardinalidade
Created by Professor Silvana

Table/Note
Add Table(T) Edit
Add Note(N) Remove
Align All Clear All
Keys

Categoria	Valor
Falhou (0%-20%)	1
Fraco (21%-49%)	2
Suficiente (50%-69%)	3
Bom (70%-89%)	4
Excelente (90%-100%)	5

Alterar Nota Nota=4 (70% - 89%) - Bom Máx.Pontos:5.00

Histórico de Respostas
Q.268
4.00
M.P.
4.00

Figura 5-9. Janela de execução de exercício DER - Perfil do professor.

A janela de execução de exercícios adapta-se também às questões de texto livre e de escolha múltipla, e têm outras funcionalidades diversas associadas que são descritas no manual do sistema.

5.1.3.2 Perfil do aluno

As funcionalidades do sistema do perfil do aluno estão agrupadas em três módulos, que atendem às necessidades dos alunos, elas são:

1. Gerir o Aluno: módulo com as funcionalidades que permite gerir o perfil editando informações pessoais, redefinir a instância da BD e monitorizar o log de acesso registado pela ferramenta.
2. Acompanhar a avaliação: módulo com um conjunto de funcionalidades para visualizar as atividades e responder às questões de escolha múltipla, de texto livre,

SQL e DER, bem como aceder aos relatórios para verificar o desempenho e acompanhar o progresso em cada atividade.

3. Explorar o Conteúdo: este módulo oferece um conjunto de funcionalidades para os alunos visualizarem o livro digital do LabDER e exportarem o material didático em formato PDF.

Outras funções do painel principal do aluno (Figura 5-10): DER-Livre, SQL-Livre e Ver conteúdo. Cada aluno tem uma instância de BD MySQL, que é carregada com um conjunto de esquemas de dados para ser usada com o material didático. O SQL-Livre fornece o acesso à instância da BD, através de uma interface simples que devolve o resultado de qualquer script SQL submetido àquela. O DER-Livre é uma interface para projetar qualquer DER, e foi disponibilizado de forma a que os alunos pudessem seguir as instruções do professor durante a aula ou usar conforme a necessidade. Esta interface tem mais de 59 funcionalidades que associam a combinação de teclas de atalho e funções do rato para diminuir o tempo de construção dos diagramas ER. O Ver conteúdo permite ao aluno aceder ao livro digital.

O painel do menu principal no perfil do aluno fornece acesso a todas as funcionalidades projetadas para o aluno. A partir dele, o aluno consegue visualizar as avaliações ativas, acompanhar o desempenho através dos relatórios, aceder aos recursos disponibilizados pelo professor, desenhar digrama ER e executar scripts SQL.

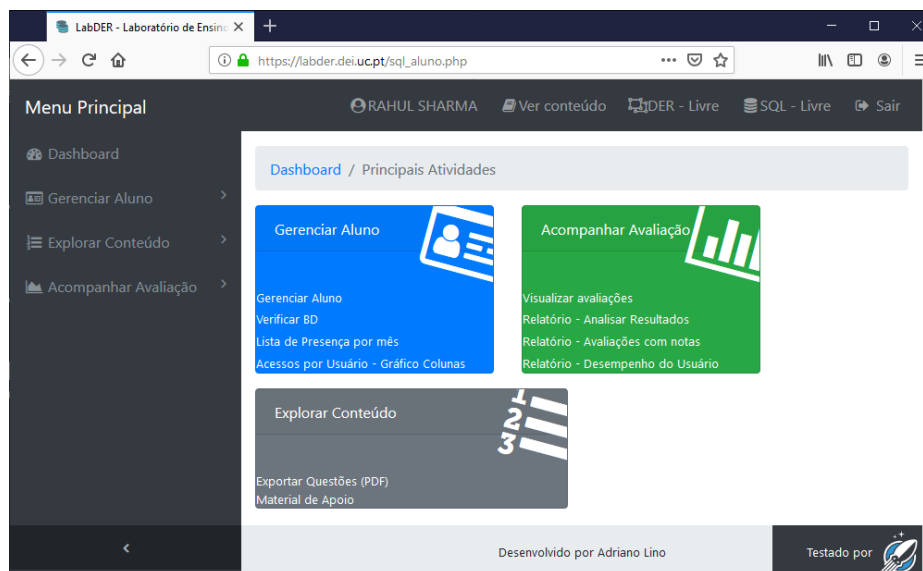


Figura 5-10. Janela menu principal - Perfil do aluno.

A janela desenvolvida para o aluno visualizar o livro digital pode ser vista na Figura 5-11. Qualquer alteração ao conteúdo do livro didático realizada pelo professor será refletida neste painel para todos os alunos.

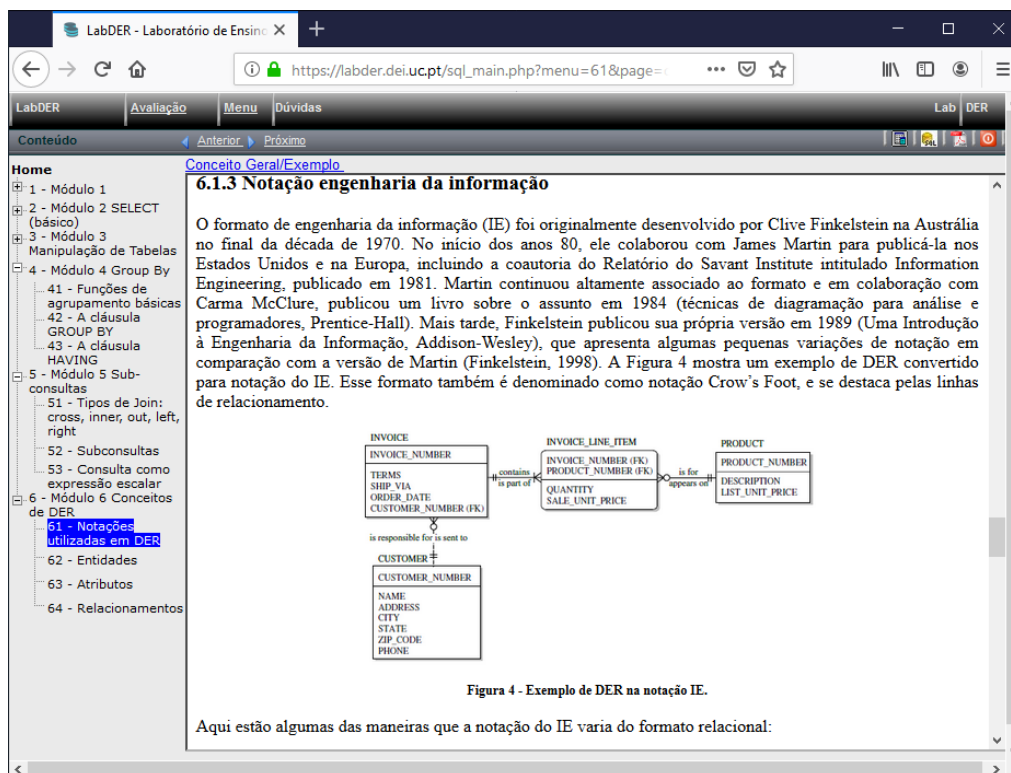


Figura 5-11. Janela para visualizar o livro digital - Perfil do aluno.

Um resumo dos tópicos abordados no livro digital pode ser visto na Tabela 5-5. Os capítulos do livro digital são acessíveis pelo menu lateral da janela - visualizar livro, que também podem ser exportados para o formato PDF. No total, o livro digital tem 104 páginas, sendo 40 do capítulo de conceitos do modelo ER.

Tabela 5-5. Tópicos abordados no livro digital do LabDER.

Capítulos	Tópicos abordados
Cap. I	Conceitos de BD, SGBD, Álgebra Relacional, Visão Geral de SQL.
Cap. II	Elementos de Sintaxe SQL, Criação de Sintaxe BD, Consultas, Restrições, Ordenação e Operadores Lógicos e Aritméticos.
Cap. III	Instruções DDL e DCL, Criar / Eliminar Objetos, Inserir / Atualizar / Excluir Instruções, Sintaxe de Restrições, Índices, Visualizações, Procedimentos, Gatilhos.
Cap. IV	Instruções SQL com GROUP BY, consultas complexas com HAVING.
Cap. V	Junções: cross, inner, out, left, right join. Subconsultas: numa expressão de coluna, na cláusula WHERE como Critério de Filtro, Cláusula FROM como Tabela, Cláusula HAVING como Seletor de Grupo, Consultas Complexas Misturando Todos os Módulos SQL.
Cap. VI	Conceitos do Modelo ER: O Modelo de Bases de Dados Relacionais, Modelo de Relacionamento de Entidade (ER), Modelação Avançada de Dados, Normalização de Base de Dados.

A janela de execução de exercício contém diversas funcionalidades, entre elas a de se adaptar a qualquer tipo de questões como SQL, DER, escolha múltipla e questão de texto livre. Todas as respostas do aluno são interpretadas por essa janela como uma inserção de resposta na BD. A função de histórico de respostas foi desenvolvida para ajudar o aluno a desenvolver soluções mais rapidamente, uma vez que recuperam a resposta submetida que pode ser alterada

e submetida novamente. Outros recursos adicionais estão descritos no manual do sistema como ver exemplo de resultado da resposta do professor ou visualizar a figura do esquema associado à questão. A Figura 5-12 ilustra a janela de execução de exercícios na visão do aluno adaptada para o tipo de questão SQL, contendo a resposta submetida pelo aluno, o *feedback* semântico e o histórico de respostas.

LabDER - Laboratório de Ensino e Aprendizagem de DER - Exercícios - Google Chro...
 https://labder.dei.uc.pt/sql_exercicio.php?cdProb=519&cdResposta=73&idaval=3348..

Questão 518 519 520 521 522 523

Prática SQL Início:29/05/2019 07:00:00 - Término:11/06/2019 22:00:00

522-Quais os associados provisórios envolvidos em mais do que 3 Projetos?

```

1 select nomep, associado.socioid,
2     count(projid) as 'envolvido em projetos'
3     from associado
4     join trabalhaem on trabalhaem.socioid=associado.socioid
5     group by nomep, associado.socioid
6     having count(projid)>3
  
```

Enviar Limpar

OBS: Seu SQL está 88,96% correto.
 Tente novamente para alcançar a solução ótima.

```

select nomep, associado.socioid, count(projid) as
'envolvido em projetos' from associado join
trabalhaem on
trabalhaem.socioid=associado.socioid group by
nomep, associado.socioid having count(projid)>3
  
```

nomep	socioid	envolvido em projetos
Ana	1	5
Joana	2	4

2 row(s)

Histórico de Respostas

Q.518	Q.519	Q.520	Q.521	Q.522	Q.523
0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	86.85	88.96	84.25
0.00	0.00	0.00	87.01	87.01	84.25
0.00	0.00	81.33	0.00		84.25
0.00	0.00		85.97		
0.00	88.58		87.01		
85.09			87.01		
85.09			87.07		
MP	MP	MP	MP	MP	MP
85.09	88.58	81.33	87.07	88.96	84.25

Figura 5-12. Janela de execução de exercício SQL - Perfil do aluno.

A janela de execução de exercício de DER (ver Figura 5-13), habilita também o aluno a desenvolver novas soluções de diagramas ER a partir de respostas anteriores. No caso de questões muito trabalhosas que envolvam muitas tabelas, o aluno pode salvar e depois clicar no histórico para recuperar e continuar a tentar chegar à solução.

270-Identifique e desenhe o ER dos componentes da tabela mostrada na Figura Q3.17, usando a terminologia correta. Use seu conhecimento das convenções de nomenclatura para identificar e desenhar as prováveis da tabela (s) e chave(s) estrangeira(s).

```

    graph LR
      Job[Job] --> Employee[Employee]
      Employee --> Dept[Dept]
      Job --- JobCode[Job_Code]
      Job --- JobName[Job_name]
      Employee --- EmpNum[Emp_Num]
      Employee --- EmpLname[Emp_Lname]
      Employee --- EmpInitial[Emp_Initial]
      Employee --- EmpFname[Emp_Fname]
      Employee --- DeptCodeDept[Dept_Code_Dept]
      Employee --- JobCodeJob[Job_Code_Job]
      Dept --- DeptCode[Dept_Code]
      JobCode --- JobCodePK[INTEGER PK]
      JobName --- JobNamePK[CHAR(20)]
      EmpNum --- EmpNumPK[INTEGER(5) PK]
      EmpLname --- EmpLnamePK[CHAR(20)]
      EmpInitial --- EmpInitialPK[CHAR(1)]
      EmpFname --- EmpFnamePK[CHAR(20)]
      DeptCodeDept --- DeptCodeDeptFK[CHAR(5) FK]
      JobCodeJob --- JobCodeJobFK[INTEGER FK]
      DeptCode --- DeptCodePK[CHAR(5) PK]
  
```

Faltou o campo nome no departamento. Altere e depois tente outra vez para conseguir Excelente.

Created by Professor

Q.268	Q.269	Q.270	Q.271	Q.272	Q.273	Q.274
5.00	1.00	3.00	2.00	2.00	5.00	2.00
5.00	1.00	3.00	2.00	3.00		5.00
2.00	3.00	3.00	3.00	3.00		5.00
4.00	4.00	3.00	5.00	3.00		
3.00	4.00	4.00		3.00		
1.00	4.00	4.00		3.00		
5.00	4.00	5.00		3.00		

O professor atribuiu a avaliação: Nota=4 (70% - 89%) - Bom

Figura 5-13. Janela de execução de exercício DER - Perfil do aluno.

A janela de execução de exercício DER foi desenvolvida para aceitar a combinação de três tipos de exercícios: (1) o tipo de transcrição, (2) o tipo de cenário, e (3) o tipo de normalização. O LabDER suporta essas três formas elementares de serem criados os exercícios, e o painel de execução de exercícios encarrega-se de apresentar ao aluno, tal como ao professor, esses elementos. Vale a pena lembrar que esses três tipos de questões podem e devem ser combinados, e assim favorecer uma interpretação mais rápida da questão pelo aluno.

Após a implementação do ambiente, seguiu-se para a fase de teste, em conformidade com o processo de desenvolvimento.

5.2 Teste do LabDER

Este capítulo tem como objetivo descrever as fases de testes do ambiente LabDER que envolve muitas atividades que são conduzidas de maneira diferente, visando objetivos diferentes. Cada tipo de teste tem um propósito e está alinhado com os objetivos associados ao atendimento de requisitos e melhoria da qualidade do ambiente LabDER.

Para verificar e validar o LabDER foram executados testes de sistema que são testes relacionados com o comportamento do sistema como um todo, a partir da validação dos requisitos funcionais e não funcionais, conforme definido pelo intuito do sistema (Myers, Sandler, & Badgett, 2012). Os testes de sistema incluem testes funcionais, de desempenho e de usabilidade. Todos os testes seguiram o mesmo processo de teste e foram executados em ambiente de teste especialmente configurado para simular o ambiente do utilizador final da forma mais realista possível.

Após os testes, o sistema está apto a ser utilizado em sala de aula com capacidade até 1.000 alunos, sendo o acesso feito simultaneamente por alunos e professores durante as aulas e as avaliações da cadeira de bases de dados.

5.2.1 Tipos de Teste do LabDER

De acordo com Myers et al. (2012), “O teste é o processo de executar um programa com a intenção de encontrar erros.” (Myers et al., 2012). Para Burnstein (2013) o teste de software “é o processo de exercitar um componente de software usando um conjunto selecionado de casos de teste, com a intenção de (a) revelar defeitos e (b) avaliar a qualidade” (Burnstein, 2003). A combinação dessas definições explica a natureza do teste de software.

O teste de software é um passo importante e essencial no processo de desenvolvimento de software que determina se um produto funciona de maneira correta e eficiente de acordo com os requisitos. A necessidade de provar que o software é capaz de executar com exatidão aquilo que a foi proposto, e criar um conjunto de evidências para atestar a sua correção, auxiliou no planeamento dos diferentes tipos de testes de software.

Os tipos de testes aplicados no LabDER foram definidos de acordo com os objetivos que se pretendiam alcançar com o uso do ambiente de trabalho.

O primeiro objetivo era atender aos requisitos funcionais do LabDER. Assegurar que todas as funcionalidades são executadas conforme especificado, principalmente tendo em consideração que o LabDER é uma nova versão dum sistema já existente (LabSQL) e que

herda diversas funcionalidades que se devem manter em conformidade com a versão anterior. Neste caso é indicado o teste de funcionalidade para atender a esse objetivo, que é uma técnica que se baseia na especificação de requisitos do ambiente de trabalho e tenta encontrar discrepâncias entre a execução do ambiente de trabalho e o que está descrito na especificação de requisitos (Myers et al., 2012).

O segundo objetivo era garantir o acesso simultâneo com mais de 500 alunos durante a resolução de listas de exercícios e avaliações no LabDER. O teste de performance tem o objetivo de atestar uma atuação adequada de utilização do ambiente de trabalho, sem riscos de bloqueios ou quedas que possam levar ao desincentivo dos alunos e descrédito dos professores.

E, por fim, e não menos importante, foi aplicado o teste de usabilidade para encontrar problemas de fator humano ou de usabilidade (Myers et al., 2012). Este teste tem o objetivo de garantir que alunos e professores possam utilizar o ambiente de trabalho sem dificuldades quanto às interações homem-máquina. Uma vez que o público alvo do sistema são os alunos e professores da área de informática, é dispensada a necessidade de formações para uso do ambiente de trabalho.

A Tabela 5-6 relaciona os principais objetivos da fase de teste com os respectivos tipos de testes, bem como as suas estratégias, os testadores e o resultado esperado para cada um. Cada tipo de teste será detalhado nas secções seguintes.

Tabela 5-6. Definição do plano de testes software do LabDER.

Objetivo	Tipo de Teste	Estratégia	Testadores	Resultado esperado
1. Atender aos requisitos funcionais e não funcionais do LabDER	Funcionalidade	Manualmente (Caso de Teste)	5 testadores	Todas as funcionalidades corretas
2. O LabDER deve manter-se operacional com mais de 500 acessos simultâneos.	Desempenho	Automatizado	Ferramenta	Carga \geq 500 acessos Tempo de resposta: < 10 segundos
3. O LabDER deve ser dinâmico, intuitivo.	Usabilidade	Questionário aos utilizadores	Utilizador final	\geq 50 pontos na escala SUS

A partir dos resultados dos testes foi aferido se o ambiente de trabalho está apto a ser implementado e utilizado por alunos e professores, em contexto de sala de aula.

5.2.2 Processo de teste

É no processo de teste que os requisitos do sistema e os componentes do sistema são exercitados e avaliados manualmente ou recorrendo a ferramentas de automação para descobrir se o sistema está a satisfazer os requisitos especificados e se existe diferença entre os resultados esperados e os alcançados (ISO/IEC/IEEE, 2013).

O processo de teste de software representa uma estruturação de fases, atividades, artefactos, papéis e responsabilidades que têm o objetivo de padronizar as atividades, maximizar a organização e a monitorização dos testes (Silva, Machado, Viana, & Santos Silva, 2016). O principal objetivo do processo de teste é garantir que não se percam etapas críticas e as atividades sejam feitas na ordem correta (Hambling, Morgan, Samaroo, Thompson, & Williams, 2010), portanto, o processo é seguido para os três tipos de teste aplicados no LabDER: funcionalidade, performance, e usabilidade.

O processo adotado para esta investigação foi inspirado no fluxo parcial do processo padrão IEEE 829-2008, ver Figura 5-14, da documentação de teste de software (IEEE Computer Society, 2008), pois considera-se apenas a documentação para gestão do teste, uma vez que o projeto de investigação tem limitações de recursos humanos e se torna inviável a execução do fluxo completo do padrão IEEE.

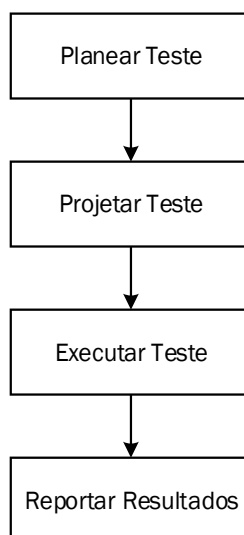


Figura 5-14. Fluxo parcial do processo padrão IEEE 829-2008. Figura adaptada de (IEEE Computer Society, 2008).

O processo de teste de software possui 4 fases que são: planejar teste, projetar teste, executar teste, e reportar resultados.

Planejar é determinar o que será testado e como isso será alcançado (Hambling et al., 2010). É a fase que determina qual o propósito dos testes e os objetivos dos testes, o que será testado, quais as técnicas que serão utilizadas além do resultado esperado para a conclusão dos testes. Os componentes do plano de teste são, a descrição do intuito, a abordagem, a identificação dos itens que estão a ser testados, os recursos a serem testados, as tarefas de teste a serem executadas, o pessoal responsável por cada tarefa, o(s) risco(s) associado(s) bem como os critérios de aprovação / reprovação dos testes (IEEE Computer Society, 2008)

Projetar o teste é a fase onde o ambiente de teste é preparado, as ferramentas de teste são definidas e são desenvolvidos todos os artefactos necessários para execução dos testes de acordo com o plano de teste. É uma fase importante, pois precisa projetar casos ou cenários de teste de forma a abranger o maior número possível de condições de teste e tornar a verificação a mais completa possível (Myers et al., 2012).

A fase de execução dos testes é a fase de execução dos procedimentos definidos e elaboração de relatórios para controle e acompanhamento dos resultados. É a execução de testes no ambiente de teste e gravação dos resultados (Ammann & Offutt, 2017). A execução de teste no ambiente de teste inclui tarefas ligadas à execução de casos de teste ou cenários de testes e permite a identificação de diferenças entre os resultados esperados e os resultados alcançados.

Reportar resultados é a fase final. A conclusão consiste em determinar se os resultados foram alcançados conforme os critérios de aprovação definidos no plano de teste e o arquivamento do material de teste. As atividades de conclusão do teste concentram-se em documentar os eventos ocorridos durante o processo de teste e que requeiram investigação, como por exemplo anomalias ou defeitos (IEEE Computer Society, 2008). Também é nesta fase que são resumidos os resultados das atividades de teste que podem providenciar avaliações e recomendações para a tomada de decisão futura sobre o software (IEEE Computer Society, 2008).

Essas fases são repetidas para cada tipo de teste atribuído à investigação. O objetivo final dos processos é o de estabelecer a confiança de que o software está pronto para cumprir o seu propósito (Sommerville, 2016). Cada tipo de teste é detalhado nas secções seguintes.

5.3 Teste de Funcionalidade

Os requisitos funcionais descrevem os serviços ou funções de um sistema, como o sistema deve reagir em determinadas entradas e como o sistema se deve comportar em determinadas situações (Sommerville, 2016). Os testes de funcionalidade são baseados na especificação funcional dos requisitos, portanto o objetivo principal dos testes funcionais é avaliar se o ambiente de trabalho faz o que deve fazer de acordo com os requisitos especificados (Dustin & Elfriede, 2002). Assim, os testes focam-se essencialmente no comportamento do sistema, tendo em conta os dados de entrada fornecidos e o resultado obtido.

5.3.1 Plano

Para garantir que o ambiente de trabalho esteja livre de defeitos, é necessário testar todos os requisitos funcionais. Inclusive as funcionalidades herdadas da versão anterior. É necessário atestar que o LabDER vai de encontro do intuito definido na análise de requisitos e o faz sem falhas.

O LabDER utiliza uma API externa para desenhar os DER e, neste caso, é sugerido um teste de interface entre o componente externo e o software para garantir que o componente responde ao que é esperado (Everett & McLeod, 2007). Uma verificação inicial, quanto às funcionalidades da API, foi executada satisfatoriamente durante o processo de escolha. Entretanto, apesar de responder satisfatoriamente individualmente, faz-se obrigatoriamente a verificação do seu comportamento ao estar integrada no ambiente de trabalho. Portanto, todas as funcionalidades da API foram incluídas nos testes de funcionalidade para garantir que o sistema, como um todo, responde, conforme os requisitos esperados.

Todas os requisitos funcionais (RF) do LabDER farão parte do processo de teste funcional, sendo que os casos de teste (TC) representam uma funcionalidade ou um conjunto de funcionalidades do sistema (Tabela 5-7 e Tabela 5-8):

Tabela 5-7. Lista de Casos de Teste (TC) por Requisitos Funcionais (RF) do perfil do professor.

Identificação RF	Identificação TC	Caso de Teste - Perfil Professor
RF 01, RF 25	TC 01	Login
RF 02	TC 02	Logout
RF 03	TC 03	Adicionar Aluno
	TC 04	Importar Aluno CSV
	TC 05	Gerir Aluno - Visualizar e Editar
RF 21	TC 06	Verificar BD
RF 04	TC 07	Adicionar Equipa
	TC 08	Gerir Equipa
	TC 09	Gerir Alunos da Equipa
RF 05	TC 10	Adicionar Avaliação
	TC 11	Gerir Avaliação
RF 09	TC 12	Agrupar Questões
	TC 13	Gerir Grupo de Questões
RF 06	TC 14	Liberar Avaliação
RF 07, RF 22, RF 23, RF 24	TC 15	Relatório - Analisar Resultados
	TC 16	Relatório - Acompanhar Avaliação
RF 07	TC 17	Relatório - Avaliações com notas
	TC 18	Frequência por mês
	TC 19	Acessos por Utilizador - Gráfico de Barras
	TC 20	Acessos por Utilizador - Gráfico de Colunas
	TC 21	Utilizadores Online
RF 08	TC 22	Adicionar Questão
	TC 23	Gerir Questão
	TC 24	Adicionar Questão de Escolha múltipla
	TC 25	Gerir Questão de Escolha múltipla
	TC 26	Exportar Questões (PDF)
RF 09	TC 27	Adicionar Grupo
	TC 28	Gerir Grupo
RF 10	TC 29	Adicionar Respostas
	TC 30	Gerir Respostas
	TC 31	Exportar Questões com Respostas (PDF)
RF 11	TC 32	Adicionar Professor
	TC 33	Gerir Professor
RF 12	TC 34	Adicionar Instituição
	TC 35	Gerir Instituição
RF 13	TC 36	Adicionar Tipo de Curso
	TC 37	Gerir Tipo de Curso
RF 14	TC 38	Adicionar Curso
	TC 39	Gerir Curso
RF 15	TC 40	Adicionar Disciplina
	TC 41	Gerir Disciplina
RF 16	TC 42	Adicionar Turma
	TC 43	Gerir Turma
	TC 44	Trocar Turma
RF 17	TC 45	Gerir Conteúdo
RF 18	TC 46	Material de Apoio
RF 19	TC 47	SQL-Livre
RF 20	TC 48	DER-Livre (TC 67 -TC 91)

Tabela 5-8. Lista de Casos de Teste (TC) por Requisitos Funcionais (RF) do perfil do aluno.

Identificação RF	Identificação TC	Caso de Teste - Perfil Aluno
RF 26, RF 38	TC 49	Login
RF 27	TC 50	Logout
RF 28	TC 51	Gerir Aluno
RF 29	TC 52	Verificar BD
RF 30	TC 53	Frequência por mês
	TC 54	Acessos por Utilizador - Gráfico de Colunas
	TC 55	Relatório - Analisar Resultados
	TC 56	Relatório - Avaliações com notas
	TC 57	Relatório - Desempenho do utilizador
RF 31	TC 58	Visualizar Avaliações
RF 32	TC 59	Exportar Questões PDF
	TC 60	Material de Apoio
RF 33	TC 61	Ver Conteúdo
RF 36, RF 37	TC 62	Responder Questões de Escolha múltipla
	TC 63	Responder Questões De texto livres
	TC 64	Responder Questões de SQL
	TC 65	Responder Questões de ER
RF 34	TC 66	SQL - Livre
RF 35	TC 67	DER-Livre (TC 67 -TC 91)

As funcionalidades foram testadas por cinco testadores, entre colegas de investigação do CISUC e colegas de trabalho no Brasil, que se disponibilizaram a colaborar nesta atividade de teste do LabDER. O número cinco foi inicialmente considerado por Nielsen e Landauer (1993) como sendo a quantidade mais que suficiente para a tarefa de teste de software, e ainda garantir que acima desse valor é um desperdício de recursos. A fórmula (1), descoberta por Nielsen e Tom Landauer (1993), representa o número de problemas de usabilidade que podem ser encontrados num teste com n utilizadores (Nielsen & Landauer, 1993).

$$N (1 - (1 - L) n) \quad (1)$$

Sendo N o número total de problemas de usabilidade no projeto e L é a proporção de problemas descobertos durante o teste de um único utilizador. O valor típico de L é de 31%, que foi encontrado em média num grande número de projetos. O gráfico de proporção de problemas encontrados pode ser visto na Figura 5-15, e é o resultado da aplicação da fórmula (1) para $L = 31\%$.

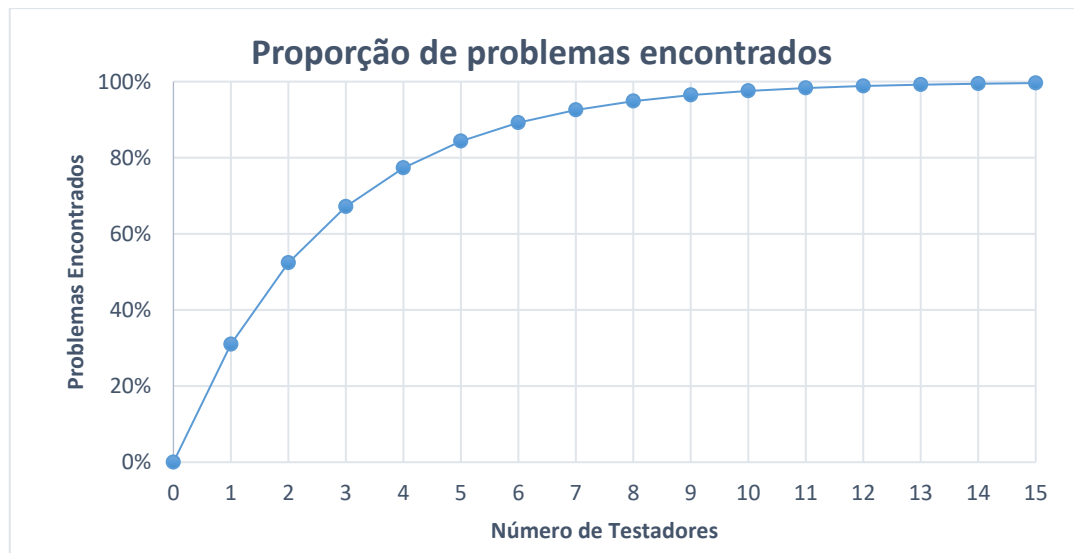


Figura 5-15. Gráfico de proporção de problemas encontrados por número de testadores. Adaptado de (Nielsen & Landauer, 1993).

O número de 5 testadores foi também comprovado ser suficiente para sistemas web (Bevan et al., 2003) . Apesar de críticas que sugerem um número maior de testadores, por exemplo entre 10 a 20 (Faulkner, 2003), optámos por manter a quantidade de 5 porque o nível de conhecimento dos utilizadores deve ser levado em consideração, que neste caso, são profissionais na área da informática.

As funcionalidades são testadas por 5 testadores num ambiente de teste preparado para este propósito, através de casos de teste que contêm o roteiro e outras informações com os passos necessários para executar e validar cada funcionalidade. Cada caso de teste é executado reiteradamente até atingir o resultado especificado e esperado.

5.3.2 Projeto

Cada funcionalidade será testada a partir de um ou mais casos de caso de teste. Um caso de teste é uma especificação das entradas, condições de execução, procedimento de teste e resultados esperados que definem um único teste a ser executado de forma a atingir um objetivo específico (ISO/IEC/IEEE, 2013). Cada caso de teste possui um conjunto de especificidades que variam de acordo com o projeto. O formato do caso de teste possui um código exclusivo de identificação, da localização que informa o caminho do objeto de teste, do título do caso de teste, da descrição resumida do objetivo do caso de teste, as pré-condições para iniciar o caso de teste, o procedimento listando os passos de execução do caso de teste, e por último a classificação se passou ou falhou observando o comportamento esperado após a execução do caso do teste (ver Tabela 5-9).

Tabela 5-9. Formato do formulário do caso de teste de funcionalidade.

Código:	Número único que identifica cada caso de teste.
Caso de Teste:	Título do teste.
Localização:	Caminho onde se localiza o objeto de teste.
Descrição:	Descrição sucinta do propósito do caso de teste.
Pré-Condição:	Uma ou mais condições que têm de se verificar para que o caso de teste possa começar.
Procedimento:	Listagem de todos os passos executados durante o teste.
Resultado Esperado:	Indica qual a saída (<i>output</i>) do sistema após a execução do teste.
Data de Teste:	Indica a data da execução do teste.
Resultado:	Indica o resultado do teste (passou ou falhou) tendo em conta o comportamento do sistema após a execução do teste.

O processo de execução, para cada caso de teste, é realizado manualmente pelo testador, e o ciclo de vida de execução do teste termina quando a funcionalidade atingir o resultado esperado, neste caso considera-se que passou. O processo é interativo, e será executado múltiplas vezes até que todas as falhas identificadas sejam corrigidas. O processo interativo para execução de cada caso de teste pode ser visto na Figura 5-16.

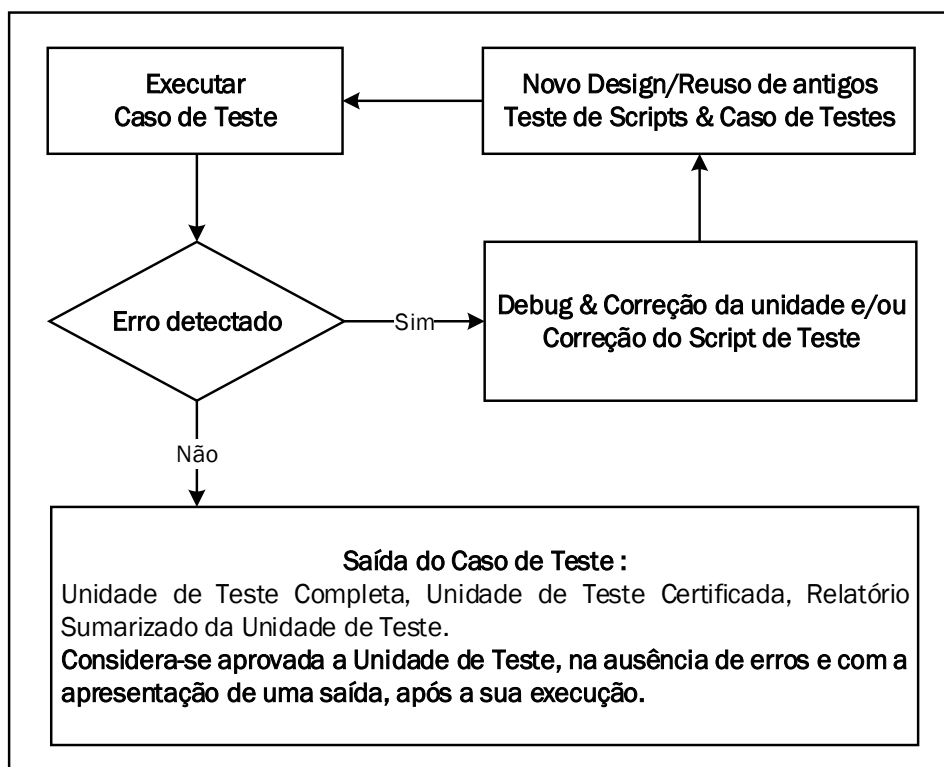


Figura 5-16. Fluxo de execução do caso de teste de funcionalidade.

Para cada funcionalidade foram planeados casos de teste, tendo em consideração critérios e procedimentos que deveriam ser seguidos nas execuções. Depois de cada execução, o bug era reportado e corrigido e novos testes eram feitos.

5.3.3 Execução e resultados

Um total de 38 funcionalidades geraram 91 casos de teste, o LabDER foi testado tendo em consideração os casos de teste do perfil do professor (ver Tabela 5-7) e do perfil do aluno (ver Tabela 5-8). Os casos de teste que falharam, foram corrigidos e os seus testes exaustivamente repetidos até obterem o resultado “aprovado”. Como exemplo, a Tabela 5-10 apresenta o caso de teste do perfil do professor TC 45 Relatório - Analisar Resultados, que obteve o resultado “aprovado” em 02 de setembro de 2018.

Tabela 5-10. Caso de teste 45 do Relatório - Analisar Resultados, perfil do professor.

Código:	TC 45
Caso de Teste:	Relatório - Analisar Resultados
Localização:	https://labder.dei.uc.pt/sql_professor.php?page=1
Descrição:	Permite ao professor visualizar um relatório com a pontuação das avaliações por aluno, alterar nota e adicionar <i>feedback</i> .
Pré-Condição:	Professor está logado no painel de Menu com as principais atividades. Uma avaliação foi aplicada.
Procedimento:	1: Selecione uma avaliação 2: Clique em "Selecionar" 3: Clicar na nota que deseja alterar 4: Selecione a questão 5: Digite a nova nota 6: Clique em "Alterar Nota" 7: Clique em "Notes" 8: Digite o <i>feedback</i> 9: Clique em OK
Resultado Esperado:	Apresenta a mensagem “Alterou a nota para:” e a nova nota. E adicionalmente o <i>feedback</i> no caso de DER.
Data de Teste:	02/Setembro
Resultado:	Passou

Os demais casos de teste do perfil do aluno estão disponíveis no Anexo C: Caso de Teste - Perfil Aluno e do perfil do professor no Anexo D: Caso de Teste - Perfil Professor, contendo a identificação, o nome do caso de teste, a localização da página, a descrição do objetivo do teste, a pré-condição do teste, o procedimento de teste, o resultado esperado, a data de teste e o resultado final do teste.

Após a conclusão de todo o processo de teste funcional, isto é, o ambiente de trabalho torna-se mais estável na execução e apto a passar pelo teste de desempenho e, por fim, pelo teste de usabilidade que contempla os testes não funcionais.

5.4 Teste de Desempenho

5.4.1 Definições

O objetivo do teste de desempenho é a validação da “rapidez” do software em relação à necessidade cotidiana de “velocidade” conforme documentado nos requisitos de software (Everett & McLeod, 2007). A velocidade, neste contexto, significa que um testador mede aspectos do tempo de resposta do software, enquanto o software está a trabalhar sob uma quantidade controlada de trabalho, chamada de carga de trabalho. Para encontrar a velocidade real do software, o testador deve executar os testes de desempenho num ambiente de trabalho controlado que se aproxime ao máximo do ambiente de produção.

O teste de desempenho é obtido através de uma série de testes que introduzem cada vez mais carga de trabalho em combinações de transações de negócio cada vez mais complexas. A velocidade do software é a resultante da combinação de tempo de resposta e carga de trabalho durante a execução dessas transações (Everett & McLeod, 2007). Além da velocidade o teste de desempenho pode medir tempos de resposta, confiabilidade, capacidade, taxa de transferência, escalabilidade e outras métricas relacionadas ao desempenho. Esses resultados podem ajudar a encontrar pontos de saturação e pontos de falha no software (Erinle, 2013).

O teste de desempenho pode ser categorizado em quatro tipos de testes conforme o seu objetivo (Meier, Farre, Bansode, Barber, & Rea, 2007b):

- 1) Teste de desempenho: Serve para determinar ou validar a velocidade, escalabilidade e / ou estabilidade. Um teste de desempenho é uma investigação técnica feita para determinar ou validar as características de resposta, velocidade, escalabilidade e / ou estabilidade do produto em teste.
- 2) Teste de carga: Serve para verificar o comportamento da aplicação sob condições normais e de carga de pico. O teste de carga é realizado para verificar se a aplicação pode dar resposta aos objetivos de desempenho desejados. Um teste de carga permite medir tempos de resposta, taxas de transferência e níveis de utilização de recursos e identificar o ponto de interrupção da sua aplicação, supondo que o ponto de interrupção ocorre abaixo da condição de pico de carga. O teste de resistência é um subconjunto do teste de carga. Um teste de resistência é um tipo de teste de desempenho focado na determinação ou validação das características de desempenho do produto em teste, quando submetido a modelos de carga de trabalho e volumes de carga previstos durante operações de produção, num

período prolongado. O teste de resistência pode ser usado para calcular o tempo médio de falha e outras métricas similares.

- 3) Teste de stress: Para determinar ou validar o comportamento de uma aplicação quando ela é enviada além das condições normais ou de carga de pico. O objetivo do teste de stress é revelar os erros da aplicação que surgem apenas sob condições de alta carga. Esses erros podem incluir problemas de sincronização, condições de corrida e vazamentos de memória. O teste de stress permite que se identifiquem os pontos fracos do aplicativo e se mostre como a aplicação se comporta sob condições extremas de carga. O teste de pico é um subconjunto do teste de stress. Um teste de pico é um tipo de teste de desempenho, focado em determinar ou validar as características de desempenho do produto, em teste, quando submetido a modelos de carga de trabalho e volumes de carga que aumentem, repetidamente, para além das operações de produção previstas por curtos períodos de tempo.
- 4) Teste de capacidade: Determinar quantos utilizadores e / ou transações um determinado sistema suportará e ainda atingirá as metas de desempenho. O teste de capacidade é realizado em conjunto com o planeamento de capacidade, que se usa para planejar o crescimento futuro, bem como um aumento da base de utilizadores ou o aumento do volume de dados. Por exemplo, para acomodar cargas futuras, é preciso saber quantos recursos adicionais (como capacidade do processador, uso de memória, capacidade de disco ou largura de banda da rede) são necessários para suportar futuros níveis de uso. O teste de capacidade ajuda a identificar uma estratégia de dimensionamento para determinar se, se deve aumentar ou diminuir a escala.

O LabDER precisa de garantir que é capaz de lidar com cargas reais esperadas do utilizador dentro de um tempo aceitável. Também é importante saber a reação esperada do ambiente de trabalho em momentos de pico, tal como as avaliações em que todos os alunos estão, ao mesmo tempo, a utilizar o sistema, e ainda descobrir o limite de utilizadores que o software irá aceitar como fator de risco, uma vez que são inaceitáveis interrupções ou falhas de sistema durante uma avaliação. Desta forma, procuramos misturar os benefícios dos conceitos de teste de performance, teste de carga, teste de stresse e de capacidade, de forma a atender aos objetivos de investigação.

Os objetivos de teste de performance, teste de carga, teste de stresse e de capacidade estão alinhados com os requisitos não funcionais (RNF 3 e 4) do LabDER, com isso, pretende-

se entregar um produto que garanta o processamento das informações num tempo adequado, bem como uma versão muito mais estável do sistema.

5.4.2 Contexto e objetivos

A descrição do cenário de uso do LabDER está relacionada com o uso do ambiente em sala de aula por alunos e professores e envolve a fase de recolha de dados que fornecerá as entradas para a fase de aprendizagem computacional. Neste cenário, serão recolhidas as respostas dos alunos em formato de DER, SQL e escolha múltipla e também será registada pelo professor, além da resposta, a avaliação para cada resposta submetida pelo aluno.

A recolha de dados teve a duração máxima prevista de um semestre letivo, período em que o professor colaborador e alunos foram convidados a utilizar o ambiente de trabalho. O professor podia criar listas de exercícios e exames, e depois avaliar conforme previsto no seu plano de ensino. Esta fase envolveu aproximadamente cerca de 500 utilizadores, distribuídos por 10 professores, com turmas entre 35 e 50 alunos, cada. O volume de dados esperado, baseado em experiências empíricas com o uso de plataformas para ensino de BD, não ultrapassaria o total de 35.000 respostas de alunos. Por exemplo, se uma turma de 50 alunos solucionar 5 listas de exercícios e 2 exames de 10 questões cada, teríamos um total de 3.500 respostas. Para atingir 35.000 respostas são necessários 10 professores, e que cada um deles forneça 7 atividades, de pelo menos 10 questões cada, para uma turma de 50 alunos.

Dentro do cenário previsto para a utilização do ambiente de trabalho, previu-se que o pico do sistema pudesse chegar a, aproximadamente, 500 utilizadores em simultâneo, situação hipotética em que os professores tivessem agendado uma avaliação para o mesmo dia e hora. A outra hipótese de pico de uso simultâneo do sistema ocorre durante as aulas práticas de laboratório, período em que o professor poderia solicitar aos alunos a leitura do material didático e a resolução de uma lista de exercícios, enquanto o professor usa o ambiente para fornecer o *feedback* aos alunos.

Os perfis de professor e aluno juntos totalizam 38 funcionalidades. Testar a performance de todas as funcionalidades do ambiente era inviável e um desperdício de recurso de tempo, uma vez que algumas funcionalidades são usadas uma única vez pelo utilizador, por exemplo alterar dados do perfil, e aceitar os termos de condições de uso do sistema, e por isso ficam fora do teste. Desta forma, surgem os cenários de teste, que representam diferentes narrativas de uso do sistema em grande escala.

O cenário de teste representa a composição das funcionalidades mais utilizadas durante o período de recolha de dados e/ou envolve um conjunto muito elevado de transações no sistema. Pode ser incluído como um teste obrigatório, um cenário de teste que simule um dia de prova, pois nesse cenário está previsto um grande volume de transações de um conjunto específico de funcionalidades. Um cenário oposto, as funcionalidades menos utilizadas, pode ser a criação de avaliações pelo professor. Por mais que seja importante a função de criar avaliações, na prática essa função seria exercida pelo professor no máximo umas sete vezes, conseqüentemente, a probabilidade de uso dessa função em paralelo é extremamente baixa e, por isso, deveria ser excluída do teste.

O cenário para o teste também considera a arquitetura do LabDER (ver subsecção 4.1.2.1) para o entendimento e planeamento dos cenários de testes. A partir da contextualização de uso do LabDER, e os objetivos de testes de desempenho descritos por (Meier et al., 2007b), foi definido um conjunto de objetivos a serem alcançados no teste de desempenho do LabDER, e os principais desafios esperados (Tabela 5-11).

Embora os potenciais benefícios superem em muito os desafios relacionados com o teste de desempenho, há a incerteza sobre a relevância dos dados resultantes com base na impossibilidade de testar todas as combinações razoáveis de variáveis, cenários e situações. Na prática, é possível reduzir ainda mais a probabilidade de falhas de desempenho catastróficas através da implementação de ações para contornar os desafios de cada teste de desempenho. Por exemplo, ao invés de usar outro ambiente de software e hardware, usa-se o ambiente de produção para eliminar o grau de incerteza dos resultados. Para este teste, foi adotada a descrição de uso geral do ambiente LabDER do ponto de vista dos utilizadores e desempenho esperado, visando extrair todas as variáveis e cenários de testes mais relevantes e garantir a abrangência das áreas vitais do uso do sistema.

Tabela 5-11. Objetivos dos testes de desempenho e desafios, adaptado de (Meier et al., 2007b).

Testes	Objetivos e benefícios do teste	Desafios e áreas não endereçados
Desempenho	<p>Determinar as características de velocidade, escalabilidade e estabilidade do aplicativo.</p> <p>Determinar se o utilizador do sistema ficará satisfeito com as características de desempenho do aplicativo.</p> <p>Suporta esforços de ajuste, planeamento de capacidade e otimização.</p>	<p>Pode não detetar alguns defeitos funcionais que só aparecem sob carga.</p> <p>Se não for cuidadosamente projetado e validado, pode ser apenas indicativo de características de desempenho num número muito pequeno de cenários de produção.</p> <p>A menos que os testes sejam conduzidos no hardware de produção, das mesmas máquinas que os utilizadores irão utilizar, haverá sempre um grau de incerteza nos resultados.</p>
Carga	<p>Determinar a taxa de transferência necessária para suportar a carga de produção máxima prevista.</p> <p>Determinar a adequação de um ambiente de hardware.</p> <p>Avaliar a adequação de um balanceador de carga.</p> <p>Detetar problemas de simultaneidade.</p> <p>Detetar erros de funcionalidade sob carga.</p> <p>Recolher dados para fins de escalabilidade e planeamento de capacidade.</p> <p>Determinar quantos utilizadores a aplicação pode manipular antes que o desempenho seja comprometido.</p> <p>Determinar quanto de carga o hardware pode aceitar antes que os limites de utilização de recursos sejam excedidos.</p>	<p>Não foi projetado para se concentrar principalmente na velocidade de resposta.</p> <p>Os resultados só devem ser utilizados para comparação com outros testes de carga relacionados.</p>
Stress	<p>Determinar se os dados podem ser corrompidos sobrecarregando o sistema.</p> <p>Fornecer uma estimativa de quanto além do carregamento de destino uma aplicação pode ir antes de causar falhas e erros além de lentidão.</p> <p>Permitir estabelecer disparadores de monitorização de aplicações para avisar sobre falhas iminentes.</p> <p>Garantir que as vulnerabilidades de segurança não sejam abertas por condições stressantes.</p> <p>Determinar os efeitos colaterais do hardware comum ou suporte de falhas no aplicativo.</p> <p>Determinar quais tipos de falhas que são mais importantes para planear.</p>	<p>Como os testes de stress são irrealistas por projeto, algumas partes interessadas podem descartar os resultados dos testes.</p> <p>Muitas vezes é difícil saber quanto stress vale a pena aplicar.</p> <p>É possível causar falhas na aplicação e / ou na rede que podem resultar numa interrupção significativa se não forem isoladas no ambiente de teste.</p>
Capacidade	<p>Permitir a realização de vários testes para comparar modelos de planeamento de capacidade e / ou previsões.</p> <p>Determinar o uso atual e a capacidade do sistema existente para auxiliar no planeamento da capacidade.</p> <p>Fornecer as tendências de uso e capacidade do sistema existente para auxiliar no planeamento de hardware.</p>	<p>Os testes de validação do modelo de capacidade são complexos para serem criados.</p> <p>Nem todos os aspetos de um modelo de planeamento de capacidade podem ser validados por testes, num momento em que esses aspetos iriam dar o maior valor.</p>

Também, foram convidados para este processo, os analistas de suporte do Departamento de Informática da UC, para providenciar suporte à infraestrutura onde está hospedado o ambiente, a fim de propor novas soluções para cada problema e limitação encontrada. A cada problema de infraestrutura encontrado, os analistas são convidados a discutir e encontrar as soluções necessárias. As interações com os analistas ocorrem somente após descartar quaisquer problemas relacionados com o código fonte do sistema.

O processo de execução para cada cenário de teste de desempenho é automatizado, utilizando uma ferramenta para execução do script que simula os principais cenários, o ciclo de vida de execução do teste termina quando a funcionalidade atingir os objetivos do resultado esperado. A partir de uma parceria com a empresa Octoperf, conseguiu-se, uma licença e formação na ferramenta para executar os testes de desempenho, com suporte técnico necessário para o desenvolvimento e implementação dos testes de desempenho (OctoPerf, 2019). O ciclo foi executado múltiplas vezes até que todas as falhas identificadas fossem corrigidas. O processo de execução do teste de desempenho é descrito nas secções subsequentes.

5.4.3 Processo de Teste de Desempenho

Todas as fases do processo de teste de desempenho requerem maior controle, pois são muitas variáveis ao serem analisadas bem como diferentes configurações dos ambientes de teste a serem inspecionados. Para este teste é necessário um processo mais detalhado e alinhado com o propósito de assegurar o desempenho do LabDER. Assim, foi-se seguindo um processo adaptado do padrão parcial da norma IEEE 829-2008 para a documentação do teste de software e sistema (IEEE Computer Society, 2008) e as 12 atividades de teste do modelo CMMI Performance (Meier, Farre, Bansode, Barber, & Rea, 2007a).

O processo segue 7 etapas (ver coluna do meio na Figura 5-17) que juntas correspondem ao detalhamento das 4 etapas propostas pelo padrão IEEE (ver coluna da esquerda na Figura 5-17) e abrangem as 12 atividades necessárias no processo de teste de desempenho de acordo com o CMMI (ver coluna da direita na Figura 5-17). Na primeira fase do processo de teste, denominada planear, estão contidas as atividades 1 e 2 do fluxo de execução do caso de teste de desempenho. Na segunda fase, denominada projetar, são descritas as atividades 3, 4 e 5. Na terceira fase, denominada executar teste, é desenvolvido a atividade 6. Na última fase do processo de teste, denominada reportar resultados, é desenvolvida a atividade 7. As atividades estão mapeadas na Figura 5-17.

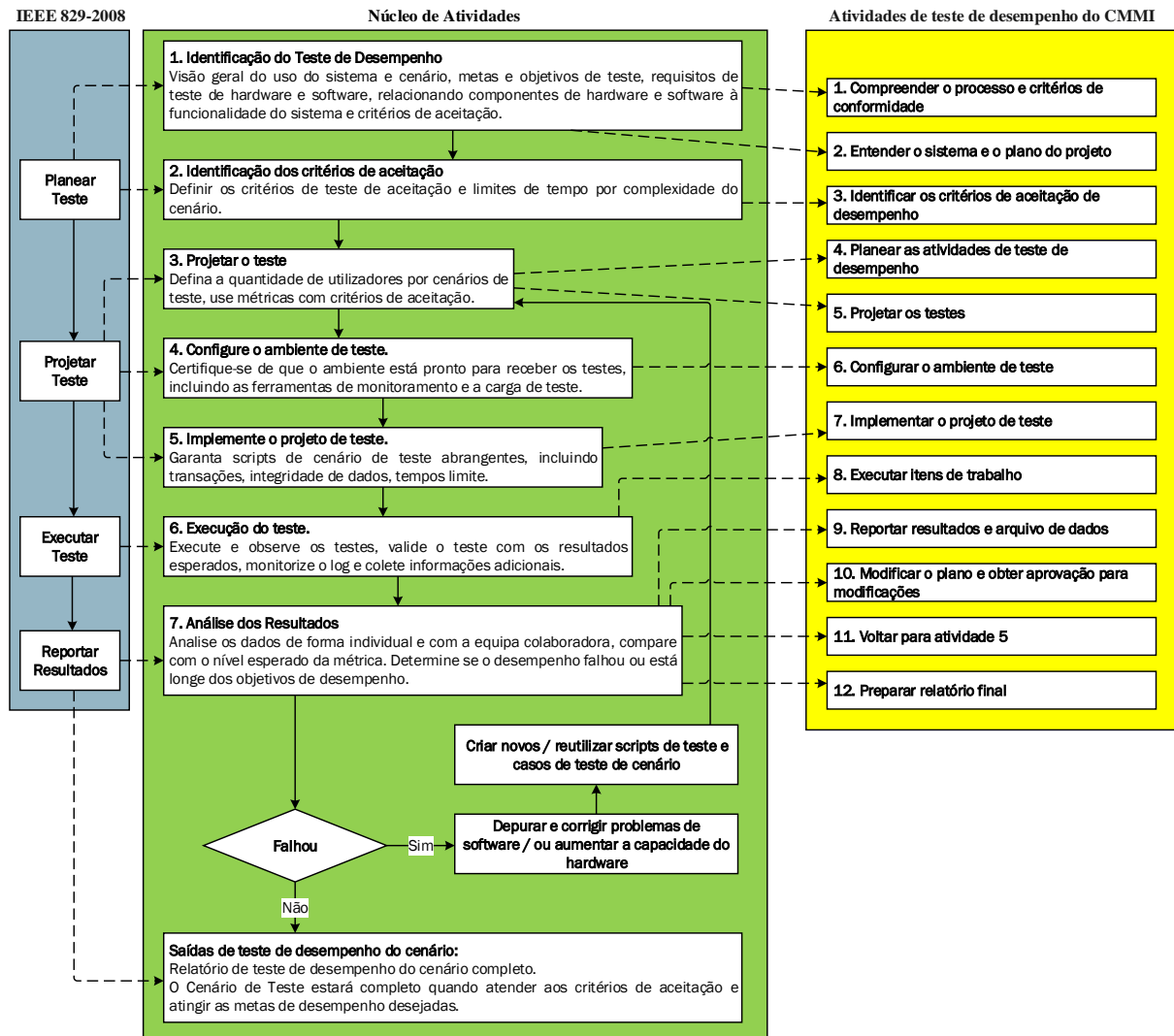


Figura 5-17. Fluxo do processo de teste de desempenho.

A primeira atividade do fluxo identifica o ambiente de teste para obter o entendimento completo de todo o ambiente de teste. Essencialmente descreve uma especificação de requisitos do sistema, incluindo o ambiente físico com as configurações de hardware, software e rede, bem como as ferramentas e recursos disponíveis para teste. Esse processo é revisto periodicamente ao longo do ciclo de vida do projeto e alterado, caso necessário.

A segunda atividade, é a identificação dos critérios de aceitação de desempenho, através da identificação de metas e restrições de tempo de resposta. É desejável, encontrar os critérios de sucesso do projeto que podem não ser detetados por essas metas e restrições; por exemplo, recorrendo aos testes de desempenho para avaliar qual a combinação de configurações que resultará nas características de desempenho mais desejáveis.

Na terceira atividade, é projetar o teste com os principais cenários de teste, contendo a variabilidade dos utilizadores representativos e como simular essa variabilidade. Também são

definidos os dados de teste e as métricas a serem recolhidas. No final, essas informações são consolidadas em um ou mais cenários de uso do sistema a serem implementados, executados e analisados. Esses itens fornecerão a base para cargas de trabalho e perfis de carga de trabalho.

O cenário deve simular interações do mundo real para fornecer dados fidedignos que permitam uma tomada de decisões. Projetos de teste do mundo real permitem aumentar significativamente a relevância e a utilidade dos dados dos resultados. Para isso, é preciso determinar explicitamente os cenários de utilização mais importante, considerando as metas e os objetivos de teste de desempenho, os cenários de utilização mais comum, os cenários de utilização crítica do sistema, e os cenários de utilização intensiva de desempenho. Um projeto de teste realista produz resultados mais fidedignos e, assim, aumenta o valor dos testes de desempenho. Por outro lado, a extrapolação dos resultados de desempenho de testes irrealistas pode criar imprecisões prejudiciais, à medida que o propósito do sistema aumenta, e, portanto, estão postos de parte quaisquer desenvolvimentos de testes irrealistas.

Na quarta atividade, o ambiente de teste é configurado e preparado para receber os testes de desempenho. O objetivo é assegurar que o ambiente de teste seja instrumentado para análise e monitorização de recursos, conforme necessário. Para isso, são utilizadas ferramentas e quaisquer outros recursos necessários para executar cada estratégia, à medida que recursos e componentes se tornem disponíveis para teste. Essa atividade envolve as ferramentas de geração de carga e de monitorização de aplicações, bem como variáveis de sistema operacional, configurações de serviço web, bases de dados dedicadas, segurança e rede.

A quinta atividade, é uma instância do projeto de teste, contendo todos os cenários de teste, implementados de acordo com o projeto do teste. Os detalhes da criação de um teste de desempenho executável são extremamente específicos da ferramenta, e envolve a criação de *scripts* num único cenário de uso, que pode ser aprimorado e combinado com outros cenários para, em última análise, representar um modelo de carga de trabalho completo.

Na sexta atividade, os testes devem ser executados e feita a recolha de resultados, enquanto se monitoriza o teste e o ambiente de teste. Após a conclusão do teste, é sugerida a rápida análise dos resultados de forma a descobrir se o teste falhou. A execução fornece indicadores para validar se o ambiente de teste corresponde à configuração esperada. Ao repetir os testes, é importante certificar se a configuração se adequa às variáveis, como o tempo de espera, limpeza de cache a cada interação, número de utilizadores e tempo de execução do teste.

A sétima atividade, visa analisar os dados e relatórios individualmente e com uma equipa colaboradora. Os testes são reexecutados, sempre que necessário, até que todas as informações desejadas, sejam recolhidas. Pode considerar-se o teste do cenário específico como concluído numa configuração específica quando os valores de métrica estiverem dentro dos limites aceites, e nenhum dos limites definidos for violado. Se o teste falhar, uma atividade de diagnóstico e ajuste deve ser geralmente realizada. Caso seja corrigida alguma falha, o teste é repetido de forma a validar a correção. A repetição dos testes é importante para a obtenção de dados consolidados que providenciem conclusões.

5.4.4 Plano

O plano de teste deve assegurar que o teste ocorrerá em ambiente similar ao ambiente de produção e que é possível executá-lo de acordo com os objetivos previamente determinados. A conceção do plano de desenvolvimento de teste de desempenho deve conter as atividades 1 e 2 do fluxo de execução do teste de desempenho, que são respetivamente, a identificação do teste de desempenho e a identificação dos critérios de aceitação.

5.4.4.1 Identificação dos cenários de teste

O LabDER contém 38 funcionalidades e, portanto, é inviável e desnecessário testar o desempenho de todas as funcionalidades do ambiente de trabalho. Os critérios de seleção dos cenários de testes são: (1) funcionalidades que tenham um elevado número de acessos simultâneos, (2) funcionalidades com elevado número de transações por utilizador, (3) funcionalidades que envolvam muita complexidade para a conclusão de uma tarefa. A complexidade é determinada pelo número de componentes envolvidos na sua execução. Por exemplo, a funcionalidade Atribuir Nota do Aluno foi selecionada para o teste porque inclui a utilização de todos os componentes da arquitetura do LabDER (ver Figura 5-3). Essa funcionalidade tem uma quantidade significativa de transações e componentes envolvidos para concluir uma tarefa.

5.4.4.2 Identificação dos critérios de teste de aceitação

O tempo de espera dos utilizadores foi inspirada no trabalho de Rempel e Garret (2015), que propôs quatro categorias de desempenho de tempo de resposta para avaliar a performance de páginas web baseado numa análise empírica de sistema utilizado a nível internacional com aproximadamente 1.200 utilizadores e 800 logins em simultâneo nos períodos de picos.

Cada categoria de desempenho tem quatro elementos (ver Tabela 5-12): a definição com exemplos, um nível de tempo de resposta desejado, um nível máximo de tempo de resposta e

uma medida de estabilidade (um valor percentual a ser usado para medir o alvo) (Rempel & Garret, 2015). Os percentis são usados para medir o desempenho da página web em vez de um tempo médio de resposta para fornecer um limite melhor ao desempenho da página e suavizar quaisquer picos nos tempos de resposta medidos (IBM Tivoli, 2012; Meier et al., 2007b). Também é usado para fornecer uma métrica comparável às medições de tempo de processamento do servidor.

Tabela 5-12. Desempenho de páginas web, extraído de (Rempel & Garret, 2015).

Nome da Categoria	Tempo de Resposta Desejado	Tempo Máximo de Resposta	Estabilidade (Percentil)
Operações básicas. Exemplo: A maioria das páginas padrão ou operações simples.	< 2s	<2 s	95%
Procura complexa ou ambígua ou operações de salvaguarda. Exemplo. Principais operações de salvamento, buscas de conjuntos de resultados grandes.	< 5s	< 5s	90%
Integração ou Operações Principais de Cálculo Exemplo: Carregar Documentos, Interfaces Síncronas, Cálculos Complexos.	< 5s	< 15s	85%
Operações Pesadas Extremamente Complexas Exemplo: Operações de Cálculo e Processamento de Dados, Interfaces Intensivas de Recursos.	< 10s	< 30s	85%

Segundo Rempel e Garret (2015), esses resultados indicam as consequências não intencionais da otimização do tempo de resposta da estimulação (em oposição à otimização do tempo de resposta totalmente renderizada). Se as informações mais importantes não estiverem entre as primeiras a serem exibidas no painel, é menos provável que uma pessoa lhe dê a atenção necessária. Além disso, se informações sem importância são apresentadas antes de informações importantes, aumenta a probabilidade de se distrair e permite assim que a distração interfira na sua memória de curto prazo e na continuidade de seu processo de pensamento (Rempel & Garret, 2015).

O número de utilizadores em simultâneo utilizados por Rempel e Garret (2015), para conceber as categorias de desempenho de páginas web, aproximam-se ao número de utilizadores em simultâneo descritos no cenário geral de uso do LabDER e, por esta razão, são adotados os tempos de resposta de cada categoria como sendo os critérios de teste de aceitação do sistema.

O segundo critério de aceitação é em relação ao número de utilizadores em simultâneo. Será executado um teste de carga até 1.000 utilizadores em simultâneo. Este número foi definido com base na oferta da disciplina de bases de dados pela FCTUC. Existem cerca de 450 alunos e 10 professores dos cursos de licenciatura e mestrado em informática a trabalhar

juntos. Portanto, 1.000 utilizadores em simultâneo representam o teste de stresse, um número utilizado para projetar a escalabilidade do sistema.

5.4.5 Projeto

Nesta fase, é prevista a definição dos cenários de teste, as métricas e os seus respetivos critérios de aceitação. Também é apresentada a configuração do ambiente de teste.

5.4.5.1 Implementação dos cenários de teste e métricas

Os cenários de testes são os principais artefactos para execução dos testes. Estes são construídos considerando a especificação do ambiente e as cargas reais esperadas pelo acesso dos utilizadores ao ambiente. Se a carga de trabalho for planeada ou executada incorretamente, as medições de tempo mais precisas não terão qualquer semelhança com o tempo exibido pelo software em produção (Everett & McLeod, 2007).

As funcionalidades selecionadas para a criação dos cenários de teste do LabDER são aquelas que têm mais acesso para alunos e professores, ou que requerem um tempo de resposta aceitável para a sua conclusão. Como, por exemplo, a funcionalidade “Ver Conteúdo” que não deveria exceder o tempo de resposta da categoria operação básica proposta por Rempel e Garret (2015), pois poderia fazer com que o aluno viesse a desistir ou a abandonar o sistema após 2s. Os cenários de teste de desempenho estão descritos na Tabela 5-13, onde estão descritos os perfis de utilizador do sistema, que podem ser professor ou aluno, a identificação do caso de teste (TC), a descrição da funcionalidade a ser testada, e as configurações dos cenários de utilizadores virtuais (UVs) que são as configurações da quantidade de utilizadores em simultâneo por cenário de teste.

Tabela 5-13. Caso de teste (TC) de desempenho por perfil do utilizador de sistema e cenários de utilizadores virtuais (UV).

Perfil	TC	Descrição	UV			
			Cenário 1	Cenário 2	Cenário 3	Cenário 4
Professor	TC 01	1) Login: os professores fazem o login simultaneamente.	1	2	10	20
	TC 15	2) Analisar resultados: O professor precisa de avaliar as avaliações dos alunos, e deve garantir que o serviço permaneça estável.	1	2	10	20
	TC 04	3) Importar CSV: O professor deve atribuir uma lista de alunos dentro de um tempo de execução aceitável.	1	2	10	20
Aluno	TC 45	1) Login: Existem 1.000 alunos que podem fazer o login em simultâneo.	1	100	500	1000
	TC 86	2) DER: Os alunos vão construir diagramas, simultaneamente, durante as avaliações em sala de aula.	1	100	500	1000
	TC 85	3) SQL: Os alunos podem executar consultas SQL no sistema simultaneamente durante a avaliação e a formação.	1	100	500	1000
	TC 57	4) Visualizar Conteúdo: Os alunos podem visualizar o material didático sobre DER e SQL.	1	100	500	1000

O cenário 1 apresenta a configuração de 1 UV para todos os casos de testes dos perfis de professor e aluno, e tem o objetivo de encontrar o tempo de resposta mínimo do sistema.

O cenário 2 tem as configurações de 2 UVs para os casos de teste do perfil do professor e 100 UVs para os casos de testes do perfil do aluno. O objetivo é encontrar o desempenho do sistema, na hipótese de ocorrer uma avaliação de 2 turmas completas a aceder, simultaneamente, ao sistema.

O cenário 3 tem as configurações de 10 UVs para os casos de teste do perfil do professor e 500 UVs para os casos de testes do perfil do aluno. O objetivo é testar a carga do sistema, na hipótese de ocorrer uma avaliação para 10 turmas completas a aceder simultaneamente, ao sistema.

O cenário 4 tem o dobro da quantidade de UVs do cenário 3, e tem o objetivo de encontrar o desempenho do sistema em teste de stresse.

Quanto ao tempo de resposta, os valores de cada tempo de resposta aceitável, é definido por categoria de complexidade do teste, conforme Rempel (2015). A complexidade do teste foi associada aos cenários de teste do LabDER recorrendo à interpretação da descrição do cenário geral do uso do ambiente e da arquitetura interna. As associações dos cenários de teste com as categorias de complexidade estão descritas na Tabela 5-14. As métricas utilizadas são o tempo de resposta desejado, tempo de resposta máximo e percentil.

Tabela 5-14. Critérios aceitáveis de tempo de resposta por caso de teste do LabDER.

TC	Categoria de complexidade	Tempo de Resposta Desejado	Tempo de Resposta Máximo	Estabilidade (Percentil)
TC - Professor Login	Operações básicas	< 2s	< 2s	95%
TC - Professor Analisar resultados	Operações Pesadas Extremamente Complexas	< 10s	< 30s	85%
TC - Professor Importar CSV	Procura complexa ou ambígua ou operações de salvaguarda	< 5s	< 5s	90%
TC - Aluno Login	Operações básicas	< 2s	< 2s	95%
TC - Aluno DER	Operações Pesadas Extremamente Complexas	< 10s	< 30s	85%
TC - Aluno SQL	Operações Pesadas Extremamente Complexas	< 10s	< 30s	85%
TC - Aluno Visualizar Conteúdo	Integração ou Operações Principais de Cálculo	< 5s	< 15s	85%

Os sete cenários descritos na Tabela 5-14 são representações realistas do uso do LabDER por professores e alunos. O tempo de resposta desejado é a meta desejada do sistema, podendo ser aceite, no pior dos casos, o tempo máximo previsto para cada cenário de teste, observando o percentil.

O cenário TC - Professor Login, faz a simulação completa do professor a fazer login seguro no sistema. Isso envolve as seguintes ações e funcionalidades do sistema: login do professor TC 01, escolher uma turma para aceder ao sistema TC 32, e a ação de sair do sistema com a funcionalidade logout do sistema TC 02.

O cenário TC - Professor Analisar resultados, faz a simulação completa de um professor a aceder a um relatório para avaliar uma resposta do aluno de uma questão do tipo ER. Isso envolve as ações e funcionalidades: login do professor TC 01, escolher uma turma TC 32, depois a seleção do relatório analisar resultados TC 15, depois as ações de selecionar e confirmar a avaliação, em seguida selecionando o primeiro aluno e sua resposta ER, e depois a atribuir um valor à resposta do aluno, e finaliza o sistema fechando a janela e com logout do sistema TC 02.

O cenário TC - Professor Importar CSV, faz a simulação completa de um professor a importar dados de alunos para o sistema utilizando um arquivo CSV, isso envolve as ações de login do sistema TC 01, da seleção da funcionalidade de importar alunos TC 04, em seguida a ação de upload do arquivo CSV e a confirmação de importação do arquivo, e finaliza a saída do sistema com o logout do professor TC 02.

O cenário TC - Aluno Login, simula o acesso seguro do login do aluno no sistema, que envolve as ações e funcionalidade de login TC 45, inserindo as credenciais, em seguida visualiza o primeiro painel do sistema que mostra as avaliações disponíveis TC 54, e por fim sai do sistema utilizando a funcionalidade logout TC 46.

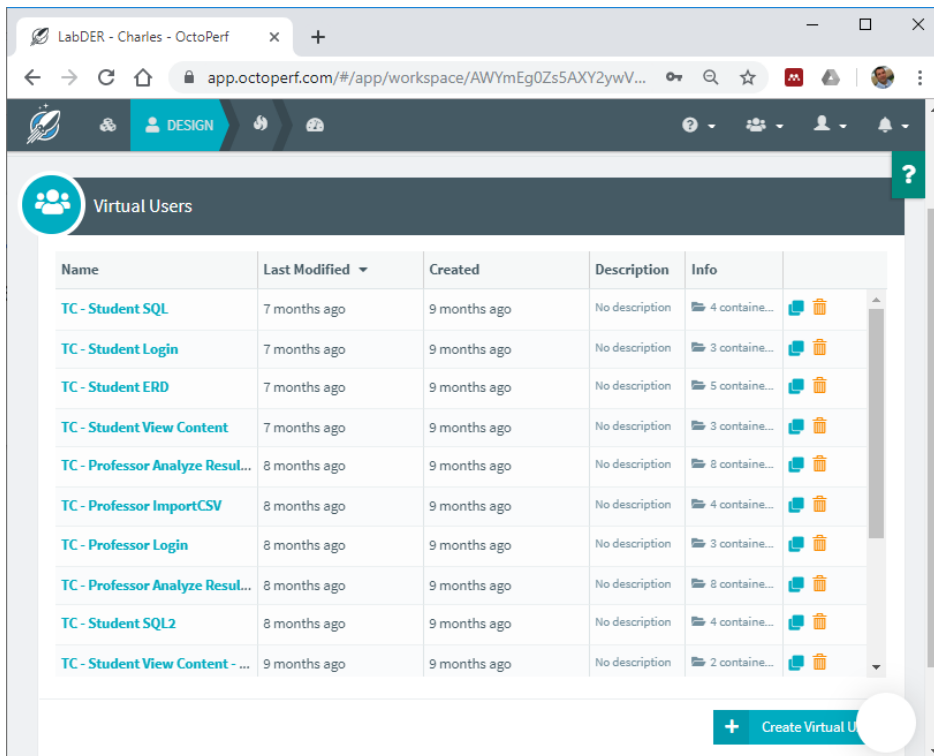
O cenário TC - Aluno DER, faz uma simulação completa do aluno a responder a uma questão de diagrama ER de entre uma lista de questões disponíveis na avaliação, envolve as ações e funcionalidades: do aluno ao fazer login (TC 45), que em seguida apresenta um painel da lista de avaliações disponíveis (TC 54), pela qual acede à avaliação e em seguida à questão do tipo ER a ser respondida. Posteriormente submete a resposta de DER (TC 86), e, por fim, o cenário é concluído com a ação de fechar a janela e efetuar o logout do sistema (TC 46).

O cenário TC - Aluno SQL, faz a simulação completa de um aluno a responder a uma questão do tipo SQL de entre uma lista de questões disponíveis na avaliação. Isso envolve as ações e funcionalidades: do aluno fazer login (TC 45), que em seguida apresenta um painel da lista de avaliações disponíveis (TC 54), pela qual acede à avaliação e em seguida à questão do tipo SQL a ser respondida. Posteriormente submete a resposta SQL (TC 85), e finaliza o cenário com a ação de fechar a janela e efetuar o logout do sistema TC 46.

O cenário TC - Aluno Visualizar Conteúdo, faz a simulação completa do aluno a visualizar um conteúdo do material didático. Envolve as ações e funcionalidades de login TC 45, e da funcionalidade de ver conteúdo TC 57, na qual o aluno tem a ação de selecionar um item de entre a lista de itens disponíveis do material didático. O cenário finaliza com o logout do aluno TC 46.

Os sete cenários descritos foram implementados na ferramenta Octoperf (ver Figura 5-18). As métricas devem medir o tempo das requisições web de cada elemento interno dos cenários, e serão agrupadas por funcionalidades para fornecerem uma informação agregada de tempo de resposta por funcionalidade, denominadas de *containers*.

O catálogo de métricas utilizadas para aferir o desempenho dos cenários de testes é fornecido no final da execução dos testes de desempenho pela ferramenta Octoperf.



Name	Last Modified	Created	Description	Info
TC - Student SQL	7 months ago	9 months ago	No description	4 containe...
TC - Student Login	7 months ago	9 months ago	No description	3 containe...
TC - Student ERD	7 months ago	9 months ago	No description	5 containe...
TC - Student View Content	7 months ago	9 months ago	No description	3 containe...
TC - Professor Analyze Resul...	8 months ago	9 months ago	No description	8 containe...
TC - Professor ImportCSV	8 months ago	9 months ago	No description	4 containe...
TC - Professor Login	8 months ago	9 months ago	No description	3 containe...
TC - Professor Analyze Resul...	8 months ago	9 months ago	No description	8 containe...
TC - Student SQL2	8 months ago	9 months ago	No description	4 containe...
TC - Student View Content - ...	9 months ago	9 months ago	No description	2 containe...

Figura 5-18. Casos de testes implementados na ferramenta Octoperf.

5.4.5.2 Configuração do ambiente

O teste foi configurado em ambiente de produção do LabDER, que inicialmente era composto por dois servidores, um de aplicação e um de BD. Foram replicadas as bases de dados no servidor de BD, onde ficam os dados institucionais como turmas, alunos, professores e as respostas dos alunos e a avaliação dos professores.

A réplica foi denominada LabDER_Performance, na qual foram inseridos os dados de 1.020 utilizadores, destes 20 sendo professores e 1.000 alunos. Para isso, foram registadas 20 turmas, em que cada professor representava uma turma de 50 alunos, à qual foi atribuída uma avaliação de 10 questões. Os dados de questões, respostas do professor, e do material de apoio, foram importadas da BD do LabDER.

As duas últimas etapas de configuração foram o registo de 50 alunos por turma e a criação de uma avaliação. Logo, foram utilizadas as funcionalidades de importar arquivo CSV e criar avaliação. A avaliação de 10 questões, distribuídas nos tipos de DER, SQL, escolha múltipla e questões de texto livre, foi registada.

Os testes eram agendados, e o servidor web foi configurado para alternar automaticamente a conexão de BD sempre que um teste era executado.

Os cenários de testes foram gravados individualmente utilizando a ferramenta Charles (disponível em <https://www.charlesproxy.com/>). O resultado de cada gravação é um script com as requisições web envolvidas no cenário.

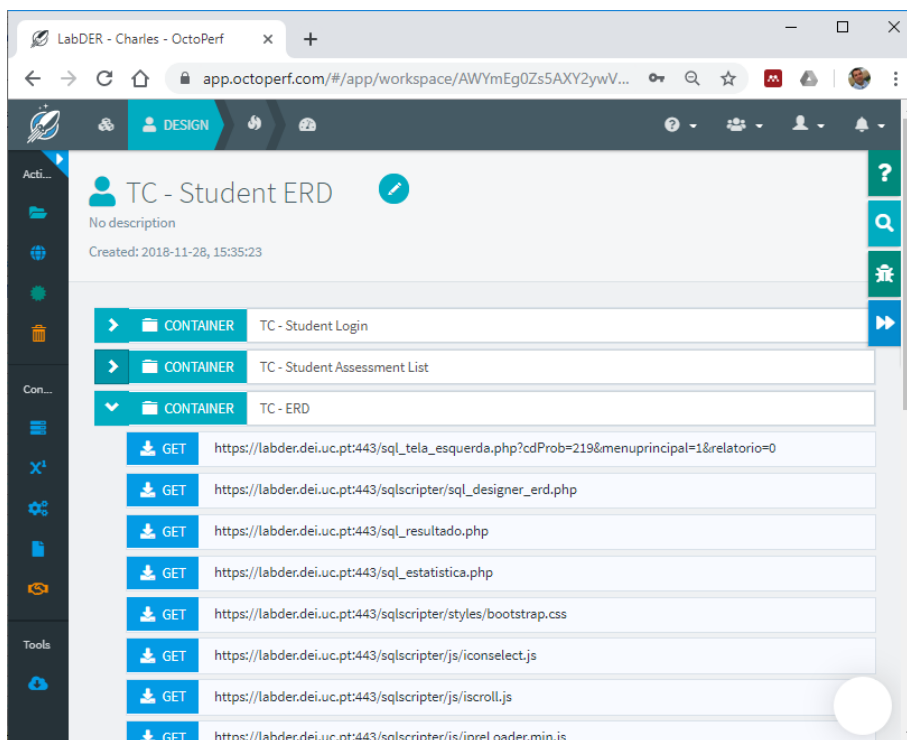


Figura 5-19. Configuração do TC - Aluno DER no Octoperf.

Os scripts de cada cenário foram importados para a ferramenta de teste de performance OctoPerf (ver Figura 5-19), juntamente com um arquivo de logins e senhas dos 1.020 utilizadores criados. Os scripts foram configurados na ferramenta OctoPerf para reagir

conforme planejado, agrupando as funcionalidades em *containers* para avaliar os desempenhos dos cenários isoladamente por funcionalidades.

A monitorização dos servidores de BD e de aplicação web eram feitos com recurso ao software Putty (disponível em <https://www.putty.org/>), que dá acesso aos logs dos servidores e permite a visualização dos testes de uma outra perspetiva durante a execução dos cenários de testes. A maioria dos testes foram conduzidos, em fins de semana, pois o sistema estava a ser utilizado, em paralelo, por alguns professores e alunos interessados no projeto.

5.4.6 Execução

Mais de 140 testes de desempenho foram realizados entre os meses de setembro de 2018 e janeiro de 2019. Como resultado, aplicaram-se diversas intervenções aos níveis da aplicação e de bases de dados e, também, no hardware. As intervenções foram agrupadas em quatro níveis (ver Figura 5-20), que representam as ações corretivas ao longo da execução reiterada do fluxo de teste.

No primeiro nível do fluxo, foram aplicadas intervenções de configuração do servidor para solucionar pequenos “bugs”. Os bugs tornam-se mais repetitivos e fáceis de identificar à medida em que aumentava o número de utilizadores virtuais, sejam bugs no código fonte ou de configuração do servidor. São exemplos de intervenções a este nível: solucionar erros no código-fonte como uma junção entre tabelas SQL, alterações nas configurações do sistema operacional dos servidores de aplicação web e BD, ajuste no nível de serviços de aplicação e BD, tamanho de cache MySQL, limite de conexão SSL, limite de conexão de BD, entre outros.

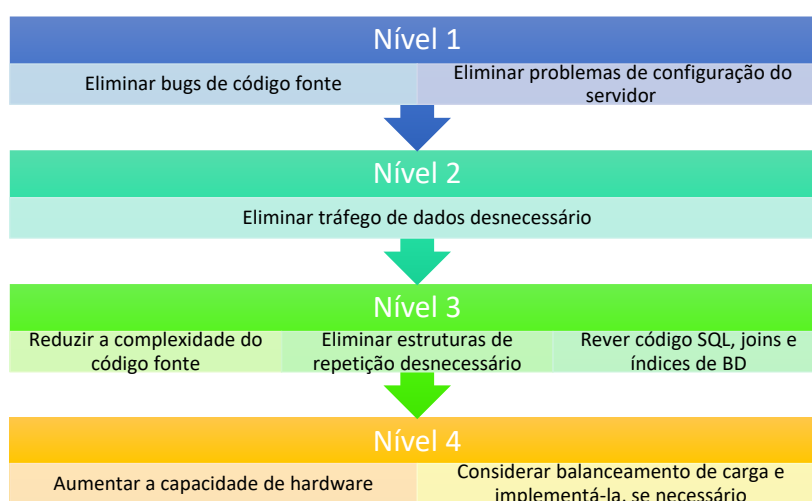


Figura 5-20. Fluxo de testes e intervenções em ambientes web.

No segundo nível, avaliou-se o tráfego de dados utilizando as métricas de taxa de transferência. Muitos scripts ou arquivos desnecessários foram revistos a fim de eliminar

tráfego de dados desnecessário, sendo aplicada até a compactação de imagens, quando necessário. Ajustes a nesse nível incluem: compressão de imagem, remover estilos de fonte, eliminar arquivos de css, javascript e todo código herdado não utilizado.

No terceiro nível, foram executadas as atividades de identificação e revisão da complexidade dos scripts em termos de desempenho, como por exemplo, eliminação de estruturas de repetição desnecessárias no código-fonte, revisão de junções em scripts SQL, adição de índices de busca na BD e até mesmo o uso de redundância de dados, se necessário. A este nível, também foram identificados os objetos passíveis de otimização, como a reescrita de relatórios em SQL, que resultou na redução de tempo de processamento do servidor.

No quarto nível, após serem esgotadas as intervenções de melhoria ao nível da aplicação e de configurações de servidor, sugerem-se alterações ao nível de hardware para melhorar o desempenho e assim atender aos objetivos do teste de desempenho. A cada alteração física foram executados novos testes de desempenho, até atingir o tempo de resposta desejado de cada cenário de teste.

Nos últimos testes, foi implementado o balanceamento de carga com dois servidores de aplicação, cada servidor web possuía a configuração de 8vCPU, 16Gb de RAM e o servidor de BD 4vCPU e 16GB, como pode ser visto na Figura 5-21. Essa configuração tem em consideração o plano de desempenho de resposta desejado, de forma satisfatória.

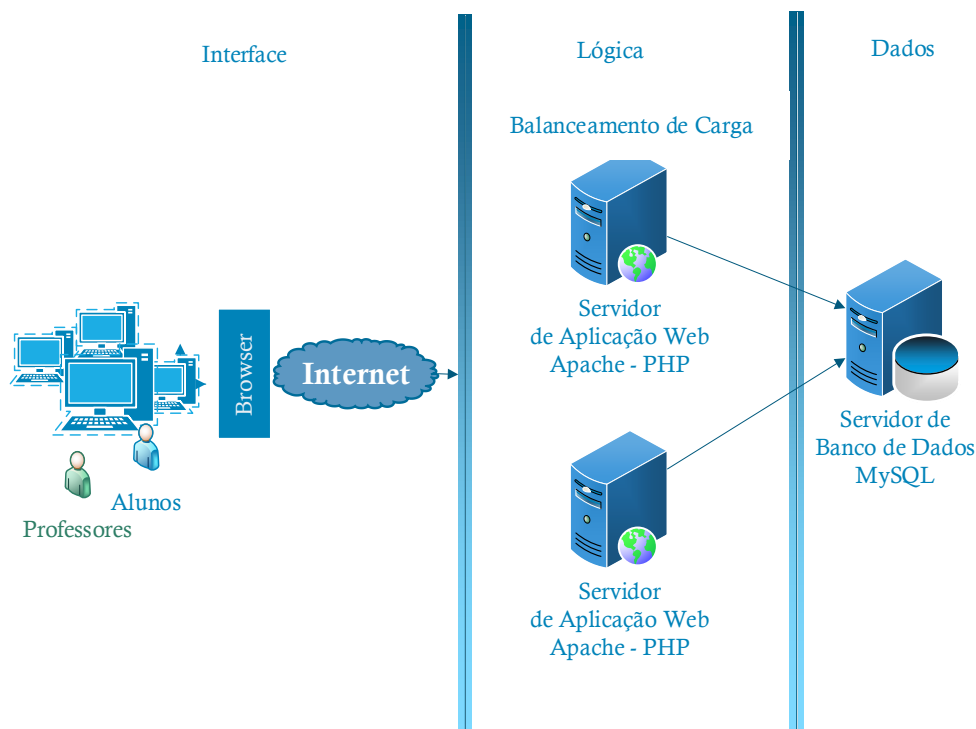


Figura 5-21. Versão final da arquitetura do ambiente LabDER.

Os relatórios de desempenho do cenário 4 são apresentados no Anexo B: Teste de Desempenho. Nessa última fase de teste não houve alterações ao código fonte, e as alterações de hardware foram suficientes para atingir a meta de desempenho estabelecida.

O caso de teste login não é apresentado individualmente porque está incluído como *container* em todos casos de teste. Portanto, no Anexo B: Teste de Desempenho, estão disponíveis os seguintes relatórios de desempenho: a) Cenário 4 - TC - Aluno DER; b) Cenário 4 - TC - Aluno Visualizar Conteúdo; c) Cenário 4 - TC - Aluno SQL; d) Cenário 4 - TC - Professor Analisar resultados; e) Cenário 4 - TC - Professor Importar CSV.

5.4.7 Análise de resultados

Os cinco casos de testes do cenário 4, com 1.000 utilizadores virtuais, fornecem um conjunto de informações importantes sobre os critérios de desempenho do sistema, pois estão relacionados com os limites de utilizadores que o sistema tem para o hardware envolvido. Portanto, é pertinente detalhar a análise dos resultados para cada um.

5.4.7.1 Teste case - Aluno DER

A Tabela 5-15 apresenta uma comparação de desempenho do cenário 4 do TC - Aluno DER com a configuração de hardware com apenas 1vCPU e a solução final com a configuração de hardware com balanceamento de carga entre 2vCPU. Os resultados do caso de teste com a configuração do hardware inicial apresentam métricas extremamente elevadas de tempo de resposta quando comparadas com a solução de 2vCPUs, sejam em tempo médio ou em percentil.

Tabela 5-15. Comparação de desempenho do cenário 4 do TC - Aluno DER.

<i>TC</i>	<i>Métrica</i>	<i>Cenário 4 - 1000 VUs Configuração 1vCPU</i>	<i>Cenário 4 - 1000 VUs Configuração 2vCPU</i>
<i>TC - Aluno DER</i>	Tempo de Resposta Médio	28,175	1,822
	Requisições	201.420	3.136.723
	Latência	9,449	1,130
	Erros	942	50
	Percentil 95%	135,626	4,411
	Percentil 90%	129,932	3,992
	Percentil 85%	52,770	3,142
	Percentil 80%	32,729	2,844

O tempo de resposta desejado para o TC - Aluno DER (ver Tabela 5-15), devido à sua complexidade é menor de 10s com percentil de 85%. O tempo obtido foi de 3,142 segundos para 85%, e pode ser considerado como aprovado. Destaca-se o impacto no número de requisições que os servidores conseguem receber com a nova configuração, passou aos 3

milhões de requisições, enquanto na configuração anterior estava limitada a 200 mil requisições.

A Figura 5-22 ilustra os gráficos dos tempos médios de resposta por *containers* do TC - Aluno DER, nas configurações de 1vCPU e 2vCPU. O *container* TC - ERD consome mais tempo de resposta, sendo 3,25s com 2vCPU e maior que 40s com 1vCPU.

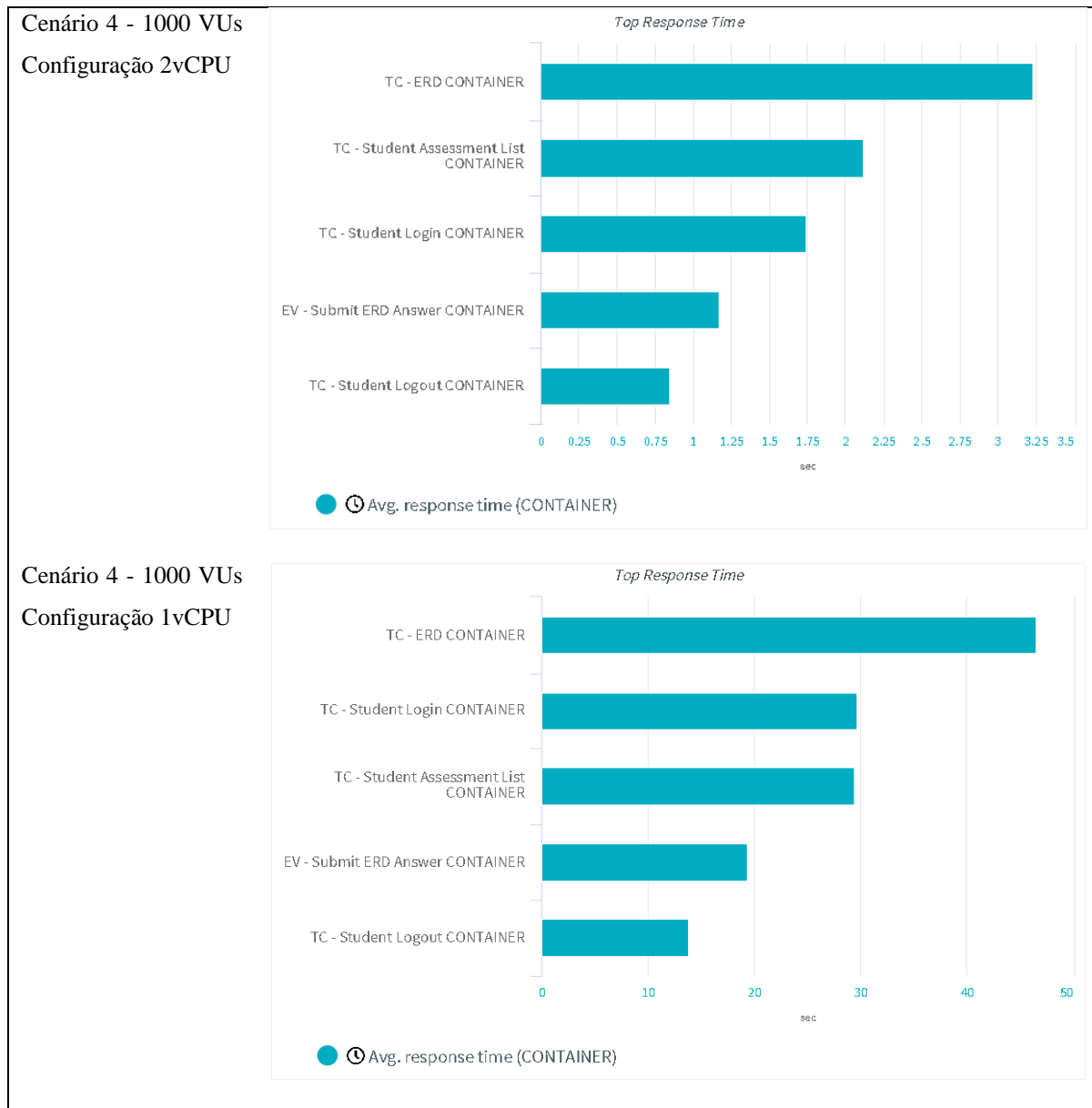


Figura 5-22. Tempo de resposta médio do TC - Aluno DER por container com 1vCPU e 2vCPU.

O balanço de carga foi aplicado após a análise do gráfico de número de requisições por tempo de resposta. Essa atividade envolveu a empresa OctoPerf e a equipa de suporte de infraestrutura da UC. A conclusão da equipa da OctoPerf foi de que existe um ponto de inflexão próximo aos 500 utilizadores em simultâneo, que ocorre em qualquer cenário de teste. Trata-

se de uma limitação da rede, o que implicava, numa fila de requisições, que o servidor não conseguisse processar, devido à sua capacidade de hardware.

No caso da configuração com 1vCPU, existe uma degradação imediata detetável visualmente pela curva ascendente do número de requisições, e conseqüentemente um aumento no tempo de resposta médio do servidor (ver Figura 5-23).

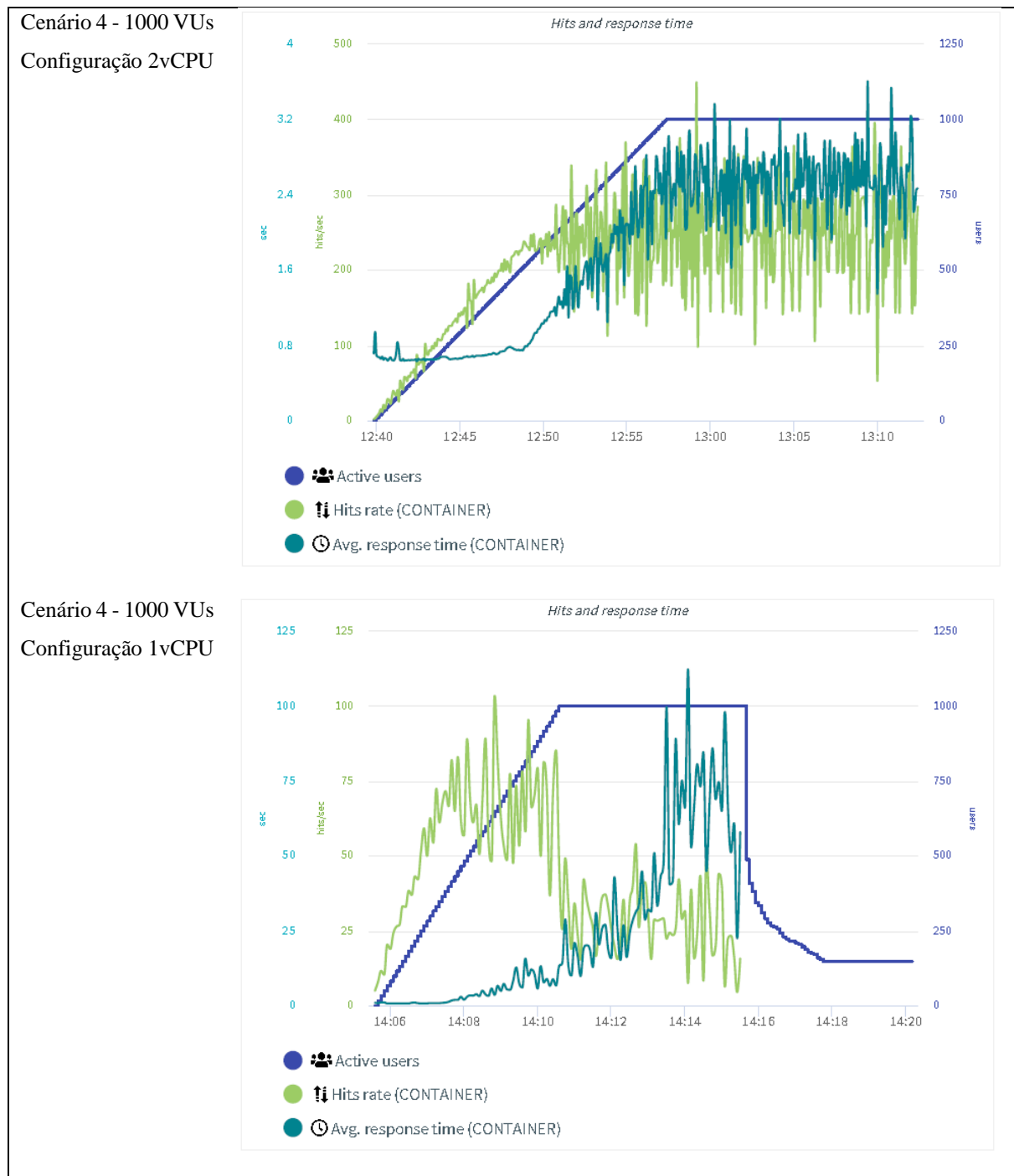


Figura 5-23. Tempo de resposta médio do TC - Aluno DER por requisições com 1vCPU e 2vCPU.

A solução de balanço de carga foi sugerida para distribuir as ligações entre os servidores, como o ponto de inflexão foi detetado próximo dos 500 utilizadores virtuais, então foi sugerido

2 servidores de aplicação para atender ao objetivo do nosso planeamento de 1.000 VU. O aumento da capacidade de hardware, de 4 para 8 núcleos, pode ser justificada pela necessidade de reduzir o tempo de processamento do servidor, que ficava sempre acima dos 95%, após 500 utilizadores em simultâneo.

O ponto de inflexão foi detetado pela análise dos tráfegos de rede externa e da rede interna. Muitos testes de software limitam-se a realizar testes na rede interna, e não consideram que, normalmente, são implementadas soluções de fibra-ótica nas redes internas, e que pode existir uma sobrecarga entre a rede interna e a externa.

No início dos testes, descobriu-se que na rede interna é possível atingir resultados excelentes para 1.000 simultâneos com 1vCPU. Inclusive, a configuração de hardware com 1vCPU de 4 núcleos foi definida a partir do teste da ferramenta do apache, sendo como mais do que suficiente para o problema proposto. Os resultados da análise do teste de desempenho da empresa OctoPerf, apontaram para a limitação de ligações e, juntamente, com as evidências partilhadas entre as equipas de informática, levaram à solução de balanceamento de carga.

5.4.7.2 Teste case - Aluno Visualizar Conteúdo

O tempo de resposta desejado para o cenário 4 do TC - Aluno Visualizar Conteúdo é menor de 5s e o tempo máximo de 15s com um percentil de 85%. Conseguiu-se alcançar um tempo de 6,071s para 85%, portanto, o caso de teste pode ser considerado como aprovado. Uma comparação de desempenho entre 1vCPU e 2vCPU pode ser visto na Tabela 5-16.

Tabela 5-16. Comparação de desempenho do cenário 4 do TC - Aluno Visualizar Conteúdo.

<i>TC</i>	<i>Métrica</i>	<i>Cenário 4 - 1000 VUs Configuração 1vCPU</i>	<i>VU Cenário 4 - 1000 VUs Configuração 2vCPU</i>
<i>TC - Aluno Visualizar Conteúdo</i>	Tempo de Resposta Médio	22,557	2,935
	Requisições	298.014	2.434.731
	Latência	6,781	1,541
	Erros	812	44
	Percentil 95%	130,398	6,582
	Percentil 90%	69,904	6,297
	Percentil 85%	21,573	6,071
	Percentil 80%	16,606	5,245

Destaca-se o impacto no número de requisições, que os servidores conseguem receber com o balanceamento de carga, que passou de, aproximadamente, 300 mil para cerca de 2 milhões e 400 mil requisições. A quantidade de 44 erros é extremamente baixa quando comparada à quantidade de 2 milhões de requisições.

O tempo de resposta médio, na configuração de 2vCPU, do container TC - Visualizar Conteúdo (ver Figura 5-24) é ligeiramente menor que o tempo de resposta desejado de 5s. Por outro lado, o servidor com 1vCPU tem o seu tempo de resposta médio para o mesmo *container* de cerca de 55s.

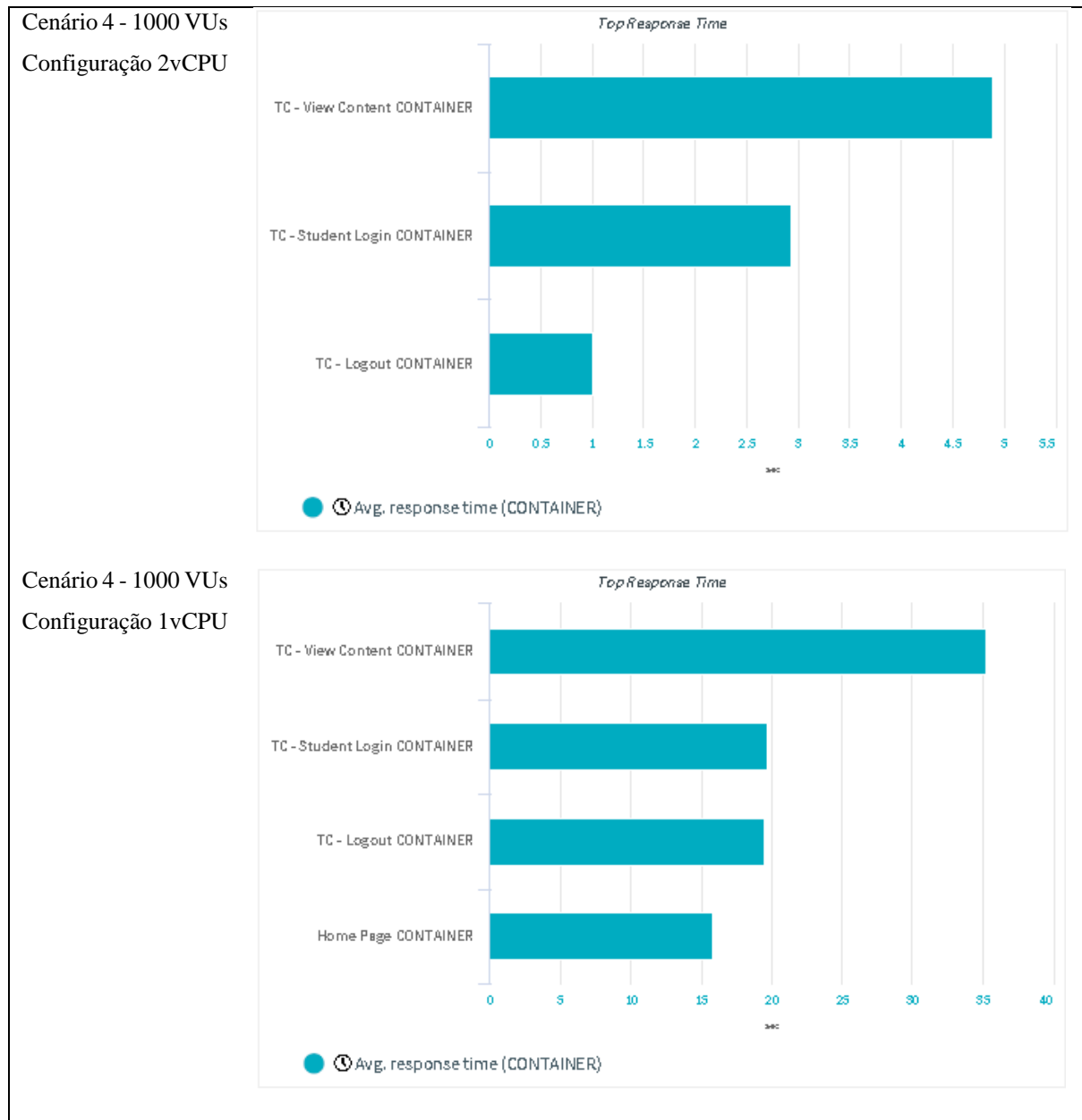


Figura 5-24. Tempo de resposta médio do TC - Aluno Visualizar Conteúdo por container 1vCPU e 2vCPU.

A Figura 5-25 ilustra o número de erros por quantidade de UVs do TC - Aluno Visualizar Conteúdo com 1vCPU e 2vCPU. No teste com 2vCPU, o número de erros é estável quando atinge o pico de 1.000 UVs, com exceção do pico igual a 1, próximo do tempo 12:10. Por outro lado, o gráfico é ascendente no teste com 1vCPU e começa a se graduar próximo dos 500 UVs, e continua a aumentar mesmo após 1.000 UVs. É possível visualizar uma curva ascendente de erros por segundo, no início os erros estão na faixa de erros entre 0 a 2 erros, por segundo;

posteriormente na faixa entre 2 a 4 erros por segundo e, por fim, começa a entrar na faixa de 4 a 6 erros, por segundo. Esse comportamento define a configuração de 1vCPU como instável.

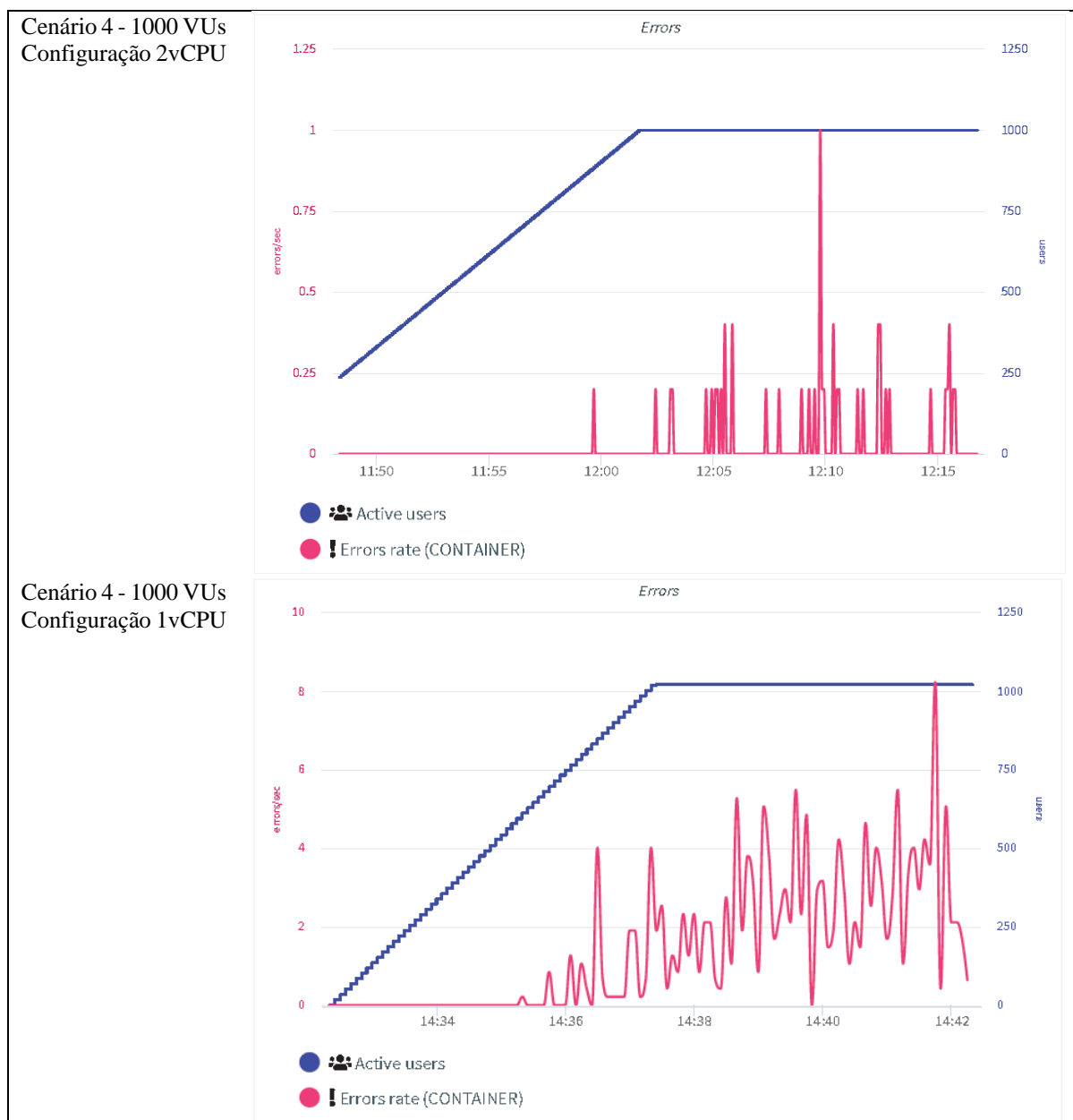


Figura 5-25. Número de erros do TC - Aluno Visualizar Conteúdo com 1vCPU e 2vCPU.

O gráfico para o caso de teste 2vCPU (ver Figura 5-25) indica a estabilidade do sistema, visível pelo padrão de erros com 1.000 UVs. Os resultados do teste de desempenho indicam que a solução de balanceamento de carga é eficaz e atende satisfatoriamente aos critérios previamente estabelecidos de tempo de resposta.

5.4.7.3 Teste case - Aluno SQL

O TC - Aluno nas configurações, é um caso de teste classificado como complexo e tem um tempo de resposta desejado menor que 10s, e tempo máximo de 30s com percentil de 85%. Na configuração de 2vCPU, o tempo foi de 10,210s para 85%, o que se pode considerar como

aprovado. A Tabela 5-17 apresenta a comparação entre os resultados dos testes nas configurações de 1vCPU e 2vCPU.

A configuração de 2vCPU, tem um resultado mais harmonioso de tempo entre os percentis 80% a 95%. Por outro lado, a configuração de 1vCPU dá uma diferença de até 10 vezes entre os percentis de 80% a 95%. Essa diferença entre tempos dos percentis é outra evidência da instabilidade da configuração de 1vCPU aceitar mais de 500 UVs.

Tabela 5-17. Comparação de desempenho do cenário 4 do TC - Aluno SQL.

<i>TC</i>	<i>Métrica</i>	<i>Cenário 4 - 1000 UVs Configuração 1vCPU</i>	<i>UV Cenário 4 - 1000 UVs Configuração 2vCPU</i>
<i>TC - Aluno SQL</i>	Tempo de Resposta Médio	13,240	4,676
	Requisições	115.534	668.501
	Latência	11,832	4,028
	Erros	699	102
	Percentil 95%	66,198	14,358
	Percentil 90%	10,036	13,495
	Percentil 85%	8,101	10,135
	Percentil 80%	6,728	7,601

5.4.7.1 Teste case - Professor Analisa Resultado

O TC - Professor Analisa Resultado, requer um tempo de resposta menor que 10s e um tempo máximo de 30s com percentil de 85%. O tempo para 2vCPU foi de 1,198s para 85%, o que se pode considerar como aprovado. A Tabela 5-18 apresenta a comparação dos testes de desempenho nas configurações de 1vCPU e 2vCPU.

O número de erros é de 0 para ambas as configurações, o que é sempre desejável num teste. O número de requisições é menor uma vez que possui no máximo 20 UVs. A instabilidade no sistema com 1vCPU pode ser vista pela discrepância de valores de percentis entre 80% e 95% (ver Tabela 5-18).

Tabela 5-18. Comparação de desempenho do cenário 4 do TC - Professor Analisa Resultado.

<i>TC</i>	<i>Métrica</i>	<i>Cenário 4 - 20 UVs Configuração 1vCPU</i>	<i>Cenário 4 - 20 UVs Configuração 2vCPU</i>
<i>TC - Professor Analisa Resultados</i>	Tempo de Resposta Médio	3,664	0,913
	Requisições	17.224	36.825
	Latência	7,431	1,035
	Erros	0	0
	Percentil 95%	24,056	4,344
	Percentil 90%	19,345	4,301
	Percentil 85%	4,697	1,198
	Percentil 80%	4,572	1,181

A Figura 5-26 ilustra o número de requisições por segundo nas configurações de 1vCPU e 2vCPU para o TC Professor Analisa Resultado.

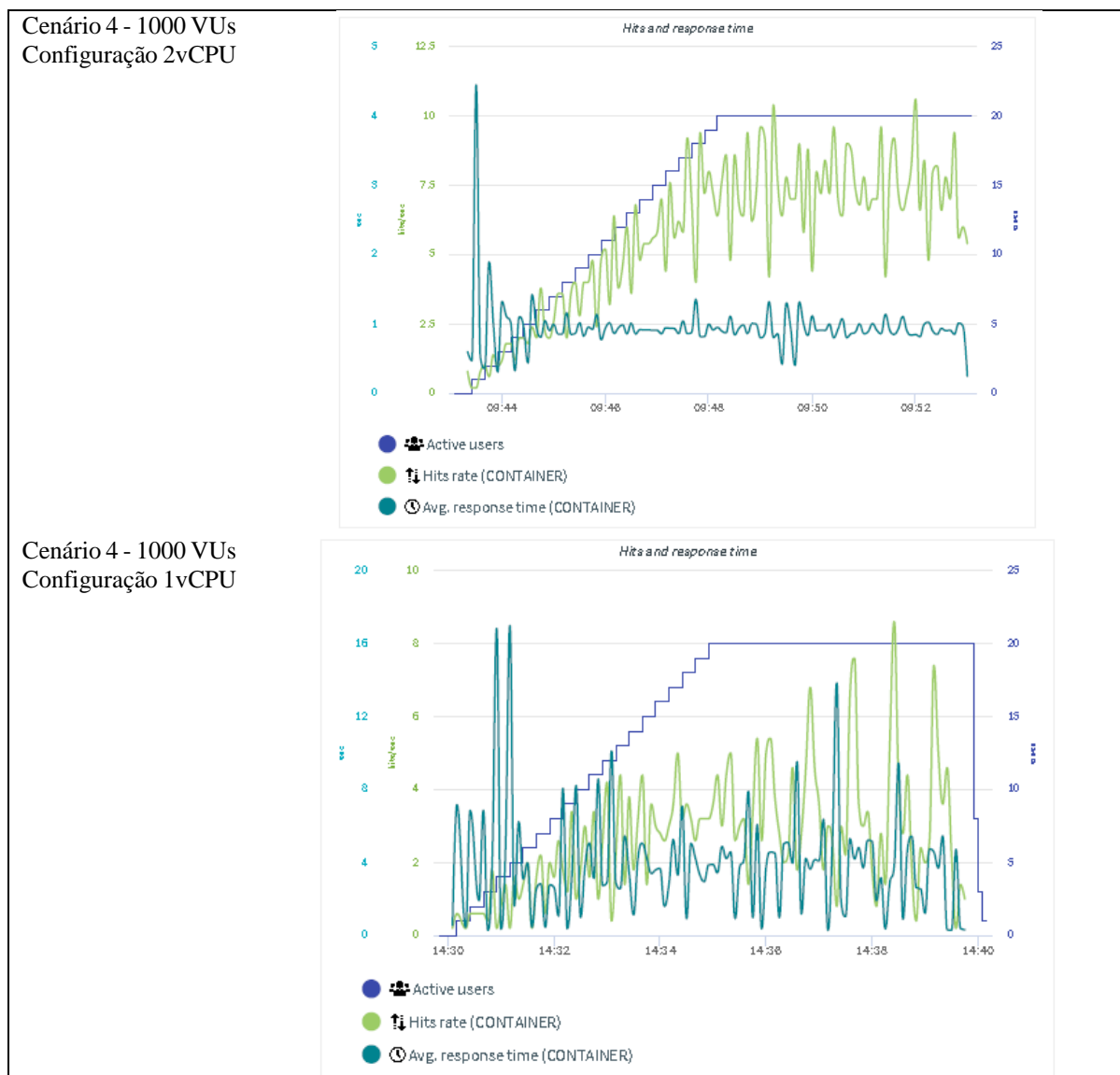


Figura 5-26. Tempo de resposta médio do cenário 4 do TC - Professor Analisa Resultado por requisições com 1vCPU e 2vCPU.

A solução 2vCPU, apresenta um gráfico onde o número de requisições acompanha o aumento da quantidade de utilizadores ao longo do tempo. Por outro lado, o teste com 1vCPU apresenta a instabilidade do sistema que não consegue responder às requisições, ao mesmo tempo que aumenta a quantidade de utilizadores. Novamente, há estabilidade no tempo médio de resposta para o teste com 2vCPU, exceto no início do teste, que tem uma carga elevada de tempo de resposta, seguido do tempo médio com um padrão sem grandes oscilações. Por outro lado, o teste com 1vCPU apresenta uma oscilação muito superior.

5.4.7.2 Teste case - Professor Importar CSV

A Tabela 5-19 apresenta uma comparação de desempenho do cenário 4 do TC - Professor Importar CSV entre as configurações de 1vCPU e 2vCPU. Ambas as configurações apresentam excelentes resultados de tempo de resposta, sempre abaixo de 1s. O tempo de resposta desejado

para este TC é de menos de 5s com um percentil de 90%, portanto as configurações atendem aos critérios de tempo.

Tabela 5-19. Comparação de desempenho do cenário 4 do TC - Professor Importar CSV.

<i>TC</i>	<i>Métrica</i>	<i>Cenário 4 - 20 UVs Configuração 1vCPU</i>	<i>VU Cenário 4 - 20 UVs Configuração 2vCPU</i>
<i>TC - Professor Importar CSV</i>	Tempo de Resposta Médio	0,610	0,444
	Requisições	5.222	11.346
	Latência	0,140	0,151
	Erros	0	0
	Percentil 95%	0,774	0,656
	Percentil 90%	0,760	0,650
	Percentil 85%	0,749	0,645
	Percentil 80%	0,744	0,642

A principal vantagem de se realizarem esses testes, foi a de encontrar um conjunto de evidências que permitiu adaptar o sistema e garantir que os requisitos, não funcionais de tempo, fossem cumpridos. Por outras palavras, pode-se garantir aos professores e alunos que o sistema aceita até 1.000 utilizadores em simultâneo, sem problemas.

Adicionalmente, foi detetada a limitação de acessos simultâneos recorrendo a testes. A limitação atual é de 500 UVs por máquina virtual. A partir do teste foi definido o ambiente de hardware, que inicialmente era de 1 unidade de vCPU com 4 núcleos e 4GB de memória, e atualmente é de 2vCPU de 8 núcleos e 16GB de memória. Sabe-se também que a tendência é de o sistema dar resposta a mais 500 UVs por máquina virtual, o que pode ser considerado como o valor de escalabilidade. As vulnerabilidades encontradas de degradação do sistema foram eliminadas durante a execução cíclica do projeto de teste, o que torna o LabDER muito mais robusto e mais leve, uma vez que foram detetados e eliminados os arquivos desnecessários.

Existem outras possibilidades de aumento do desempenho do sistema que não foram totalmente exploradas, como o balanço de carga entre BD. Entende-se como desnecessário, uma vez que a solução de balanço de carga dos servidores de aplicação garante até 1.000 utilizadores em simultâneo. É importante enfatizar que o contexto de sala de aula é muito sensível, principalmente nos momentos de avaliações formais e listas de exercícios. Qualquer falha do sistema poderia levar professores e alunos a desistir de participar no projeto. Com o resultado dos testes, e na versão atual do sistema. Pode garantir-se, a professores e alunos, que o sistema está apto a receber o dobro da capacidade de utilizadores, inicialmente planeada, em simultâneo, e que o sistema responde bem, mesmo nessas condições.

5.5 Teste de Usabilidade

Usabilidade é um conceito que pode ser empregue em qualquer ferramenta que usamos. Geralmente, coisas que são fáceis e agradáveis de usar são mais úteis do que coisas que são complicadas e cansativas de usar. Para obter uma definição melhor, podemos olhar para os padrões. Para usabilidade, o padrão é ISO 9241-11, a “Ergonomia da Interação do Sistema Humano” (ISO, 2018). Green e Pearson (2016) definem usabilidade, como está escrito na ISO 9241-11, como se segue: “A medida em que um produto pode ser usado por utilizadores específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação num contexto específico de uso” (Green David & Pearson J. Michael, 2016).

Brooke (1996) explica os três fatores de usabilidade acima mencionados da seguinte forma: Eficácia é “a capacidade dos utilizadores concluírem tarefas utilizando o sistema e a qualidade da saída dessas tarefas” (Brooke, 1996). Por outras palavras, a eficácia mede a usabilidade da ferramenta, a partir da sua saída. Por exemplo, se se pretende fazer o registo contabilístico de uma empresa (um livro de caixa, por exemplo), o Microsoft Excel é bastante utilizável em termos de eficácia, assim como, de facto, o registo contabilístico está a ser produzido.

Por outro lado, a eficiência é “o nível de recursos consumidos na execução de tarefas” (Brooke, 1996). O Microsoft Excel é mais utilizável em termos de eficiência do que um livro de caixa realizado manualmente em papel. O recurso menos usado neste exemplo é o tempo (calculadora, tinta e papel). Satisfação é a “reação subjetiva dos utilizadores à utilização do sistema” (Brooke, 1996). Todas as medidas de usabilidade dependem das capacidades e atitudes do utilizador. A satisfação do utilizador é a dimensão mais subjetiva da definição da ISO 9241-11 (ISO, 2018). Um utilizador pode, por exemplo, ter mais satisfação, ao fazer o livro de caixa, ao usar uma calculadora e / ou papel e caneta, do que fazê-lo com o Microsoft Excel. Para esse utilizador, a dimensão de satisfação da usabilidade é maior com a calculadora do que com o programa de folha de cálculo. A usabilidade geral de uma ferramenta é uma combinação desses três métodos. Um produto pode sobressair numa dimensão e ainda ter pouca usabilidade, se for executado mal nas demais dimensões.

Muitas das limitações encontradas nas ferramentas anteriores estavam relacionadas com as restrições na manipulação da interface do ambiente de trabalho. O LabDER tem como proposta ser um ambiente dinâmico que dá liberdade ao utilizador para desenvolver a aprendizagem dos tópicos relacionados a DER. Porém, é importante garantir que essas

melhorias não afetem negativamente a experiência do uso do ambiente de trabalho por alunos e professores.

A necessidade de medir a usabilidade, mesmo quando o sistema é funcionalmente executável, é corroborada por Kortum e Bangor (2013) quando afirmam que:” A medição da usabilidade tem alta utilidade porque quantifica quão bem os utilizadores podem interagir com um determinado produto ou serviço. Mesmo que um produto execute sua função técnica primária sem falhas, se um utilizador não conseguir fazer com que o produto funcione, então esse produto falhou.” (Kortum & Bangor, 2013).

Existem várias ferramentas e métodos diferentes para medir a usabilidade. Diferentes técnicas podem ser divididas em dois grupos: testes empíricos com utilizadores e inspeções de usabilidade (Riihiahho, 2000). Teste empírico do utilizador abrange todos os métodos que envolvem os utilizadores reais no processo, por outro lado, as inspeções de usabilidade devem ser feitas por especialistas desta matéria (Riihiahho, 2000).

Alguns testes empíricos de usabilidade recolhem dados por observações e geralmente são realizados para fornecer *feedback* rápido. Outros testes são mais específicos e recolhem sistematicamente dados a partir de uma série de questionários. Além disso, realizam análises estatísticas para fornecer melhores evidências de suporte aos problemas de usabilidade, em potencial.

5.5.1 Questionário SUS

System Usability Scale (SUS) é “o questionário padronizado mais amplamente utilizado para a avaliação da usabilidade percebida” (Lewis, 2018). O SUS foi desenvolvido por Brooke (1996) que defini o SUS como “o *System Usability Scale* é uma escala simples que oferece uma visão global das avaliações subjetivas da usabilidade” (Brooke, 1996).

O questionário do SUS é composto por dez afirmações, sendo cinco delas positivas e cinco negativas (Brooke, 1996). Uma mistura de afirmações positivas e negativas é geralmente usada para equilibrar várias propensões, como a tendência dos inquiridos a concordar com a maioria das afirmações, e fazer os questionados pensar em cada afirmação, antes de responder.

O SUS utiliza uma escala de *likert* de cinco pontos, variando de discordo fortemente e concordo totalmente. Os resultados são apresentados como um único número, representando uma medida composta da usabilidade geral do sistema em estudo (Lewis, 2018). Esta pontuação varia de 0 a 100, perfazendo 50 pontos na pontuação média teórica, mas a pontuação média real percebida é próxima de 70 (Lewis, 2018).

Apesar do Brooke (1996) entender e citar a definição da usabilidade pelo padrão ISO 9241-11 (usabilidade como uma combinação de eficácia, eficiência e satisfação) (ISO, 2018). Ele afirma que “as medidas usadas dentro de cada umas dessas classes de métricas podem variar amplamente” (Brooke, 1996). Na proposta, o SUS não visa avaliar sistemas em dimensões individuais de eficácia, eficiência e satisfação, como na ISO, em vez disso, pontua a experiência geral do utilizador com a combinação das três dimensões (Lewis, 2018).

De acordo com o autor “pontuações para itens individuais não são significativas por si só” (Brooke, 1996). Isso significa que a escala SUS atribui uma pontuação geral de usabilidade do sistema. O avaliador só deve olhar para esta pontuação final, e não para as pontuações sobre questões individuais. A Tabela 5-20, apresenta o questionário original do SUS, composto por dez questões que são avaliadas na escala de *likert* de cinco pontos.

Tabela 5-20. Questionário padrão do System Usability Scale (SUS).

Please rate your level of agreement with each of the following statements:	1. Strongly disagree	2	3	4	5. Strongly agree
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technician to be able to use this system.					
5. I found the various functions in this system were well integrated.					
6. I thought there was too much inconsistency in this system.					
7. I would imagine that most people would learn to use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. I needed to learn a lot of things before I could get going with this system.					

Brooke (1996) instrui o uso do questionário da seguinte forma: “Para calcular a pontuação do SUS, some primeiro as contribuições da pontuação de cada item. A contribuição da pontuação de cada item varia de 0 a 4. Para os itens 1, 3, 5, 7 e 9 a contribuição da pontuação é a posição da escala menos 1. Para os itens 2, 4, 6, 8 e 10, a contribuição é de menos 5. Multiplique a soma das pontuações por 2,5 para obter o valor global da escala de usabilidade” (Brooke, 1996). Como mencionado anteriormente, a pontuação do SUS, para um sistema, deve ser vista como um resultado da uma equação matemática que valoriza as questões pares e ímpares.

Embora Brooke (1996) aconselhe a não utilização das pontuações de declarações individuais sozinhas, pois não seriam significativas por si só, as análises fatoriais de J.R. Lewis e Sauro (2009) revelaram que os resultados das afirmações 4 e 10 juntas poderiam ser usadas para avaliar a capacidade de aprendizagem de um sistema e as afirmações 1, 2, 3, 5, 6, 7, 8 e 9

refletiriam a usabilidade (Lewis & Sauro, 2009). Essa percepção adicional pode ser muito relevante para a avaliação de um ambiente de ensino e aprendizagem, pois se for atestado que as dificuldades se centram na capacidade de aprendizagem, e não na usabilidade geral, isso poderia ser abordado com ênfase numa formação.

Bangor et al. (2008) fizeram uma extensa investigação sobre questionários prévios do SUS. Eles combinaram “quase uma década de dados do SUS” com “mais de 2.300 investigações individuais” e “200 estudos” (Bangor, Kortum, & Miller, 2008). Como resultado da sua extensa investigação, Bangor et al. (2008) produziram a seguinte ilustração (ver Figura 5-27) com base na distribuição estatística dos resultados do SUS.

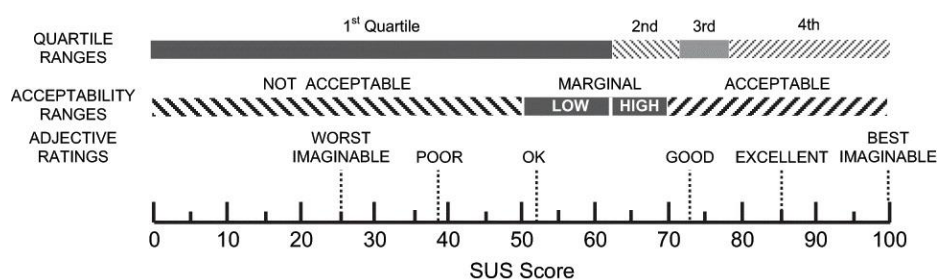


Figura 5-27. Comparação de pontuações médias por quartil, classificação dos adjetivos e aceitabilidade da pontuação geral do SUS, extraída de (Bangor et al., 2008).

Bangor et al. (2009) explicou que produtos com uma pontuação acima de 70 podem ser considerados aceitáveis. Bons produtos pontuam no SUS na faixa de 70 a 80 (Bangor, Kortum, & Miller, 2009). A Figura 5-28 ilustra como os melhores 25% dos produtos obtêm uma pontuação do SUS superior a 80, mas os “produtos realmente superiores pontuam melhor que 90” (Bangor et al., 2009). No outro lado do espectro, a figura mostra como as pontuações do SUS abaixo de 70 estão na metade inferior da usabilidade. Esses produtos com resultados de usabilidade inferiores a 70 “devem ser considerados candidatos a uma maior investigação e melhoria contínua” e “produtos com pontuações menores que 50 devem causar preocupação significativa e são considerados como inaceitáveis” (Bangor et al., 2009).

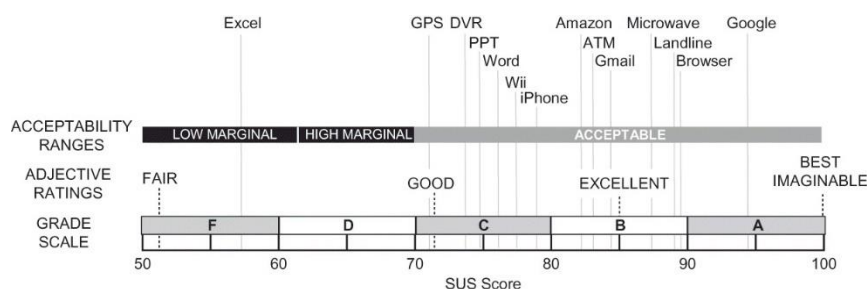


Figura 5-28. Pontuações de usabilidade para os 14 produtos mapeados nas faixas de aceitabilidade propostas por Bangor (2009). Nota. GPS = sistema de posicionamento global; DVR = gravador de vídeo digital; PPT = PowerPoint; ATM = caixa eletrônico.

Os critérios para aceitar e rejeitar a usabilidade do LabDER foram definidos a partir das definições descritas acima por Bangor (2009). Portanto, a usabilidade do LabDER é considerada como aceitável somente se obtiver uma pontuação acima de 50 pontos na escala SUS, e inaceitável se obtiver menos que 50 pontos.

O SUS tem sido amplamente utilizado desde a sua introdução devido aos seus benefícios, como: (a) facilidade em administrar o questionário tanto para o utilizador do teste como para o supervisor do teste, (b) disponibilidade gratuita e (c) as boas medidas de fiabilidade e validade (Kortum & Bangor, 2013). Lewis e Sauro (2009) citam diferentes investigações (por exemplo, (Bangor et al., 2008) sobre a fiabilidade do SUS e confirmam o que a investigação prévia sobre o SUS sugeriu ao realizar a análise fatorial em 324 questionários do SUS, que a fiabilidade do SUS, na medição da usabilidade é alta (Lewis & Sauro, 2009).

A validade do teste também pode ser aumentada realizando o teste com a maior proximidade possível das circunstâncias da vida real. Técnicas para fazer isso incluem a realização de testes de usabilidade, nas próprias instalações dos utilizadores de teste. A validade de um teste de usabilidade também pode ser aumentada recorrendo aos verdadeiros utilizadores do sistema, em comparação e como exemplo, alunos universitários recrutados para usar o programa apenas para testes (Nielsen, 1993).

5.5.2 Plano

O principal objetivo do teste de usabilidade, neste projeto, foi descobrir se os requisitos de software desenvolvidos no LabDER tinham uma taxa de aceitação satisfatória, sob o ponto de vista dos utilizadores, ou seja, ter como meta obter, no mínimo, 50 pontos na escala SUS. Para isso foi aplicado o questionário SUS, que fornece um resultado numérico que determina se a usabilidade do sistema é aceitável, acima de 50 pontos, ou não, abaixo de 50 pontos. Adicionalmente, o resultado numérico fornece interpretações do grau de satisfação dos utilizadores em escalas de adjetivação como “Justo”, “Bom”, “Excelente” e “Melhor Imaginável”, conforme descrito por (Bangor et al., 2009).

Foi prevista a usabilidade do questionário em sala de aula para as turmas que utilizaram o LabDER como ambiente de aprendizagem de conceitos de DER. O questionário foi aplicado imediatamente após uma avaliação formal, em que professores e alunos já desempenharam suas atividades no LabDER. A decisão de aplicar, em mais de uma turma, foi a de tentar garantir a fiabilidade dos resultados. Há que ter em conta que existem evidências de que alunos

tendem a avaliar negativamente a universidade, os professores e até as ferramentas, quando têm más avaliações (Baker et al., 2000; Hornstein, 2017).

O teste de usabilidade empírico é executado por utilizadores reais, portanto, é importante ter em conta fatores éticos. Pode ser muito estressante ser o utilizador de teste observado. Portanto, é planeado seguir os conselhos práticos sobre os aspetos éticos em testes com seres humanos, proposto por Nilsen (1993), como: “O teste tem como objetivo avaliar o software, não o utilizador, nenhuma informação do utilizador é compartilhada, nenhuma informação do desempenho do utilizador é compartilhada, haverá apenas um observador e todos os utilizadores de teste são tratados com respeito e de maneira neutra” (Nielsen, 1993).

Todos os participantes avaliaram o sistema imediatamente após usarem o sistema numa avaliação formal. Então, após recolher os dados, a tabela foi feita na folha de cálculo do Excel de forma a calcular as pontuações do SUS, com base nas respostas dos participantes.

5.5.3 Projeto

O projeto de teste de usabilidade do LabDER descreve onde o ambiente de teste foi aplicado, as ferramentas e todos os artefactos desenvolvidos para a execução dos testes, de acordo com o plano de teste.

O instrumento do teste é o questionário SUS, que foi escrito originalmente em inglês, depois foi traduzido para espanhol e para português. Ao todo foram 64 os participantes que avaliaram o LabDER (ver Tabela 5-21). Os alunos e professores falantes da língua portuguesa que participaram no questionário são originários do Brasil e de Portugal, respetivamente da Faculdade Pan Amazônica (FAPAN) no Brasil com 13 participantes, e Coimbra Business School (ISCAC), com 26 participantes. Os participantes que falam espanhol são da Universidade Pública da Colômbia (UNAL) na Colômbia, onde foi aplicado o questionário, com 25 participantes.

Tabela 5-21. Total de participantes do questionário SUS.

Universidade	País	Alunos	Professores
Faculdade Pan Amazônica (FAPAN)	Brasil	12	1
Universidade Pública da Colômbia (UNAL)	Colômbia	23	2
Coimbra Business School (ISCAC)	Portugal	24	2
Total de Participantes do questionário SUS:		64	

Em seguida, o questionário foi implementado na plataforma Google Forms e adicionado o seu link de acesso à plataforma do LabDER (ver Figura 5-29). Então, os utilizadores tiveram acesso ao questionário de avaliação diretamente pela plataforma.

Após implementação, foram agendados os testes com os professores de cada turma, tendo em conta o fuso horário de cada país. Os ambientes tecnológicos e de ensino foram mantidos tal e qual, como já eram usados em aula. Em todas as turmas, o fluxo de atividades foi monitorizado em tempo real via Google Analytics (disponível em <https://analytics.google.com>), e por ferramentas de conversação e e-mail, com exceção das turmas em Portugal, que ocorreram presencialmente. Como requisito do acesso ao sistema, todos os utilizadores responderam ao formulário prévio, onde aceitavam participar do teste e o investigador se comprometia a resguardar os dados conforme o RGPD nº 2016/679, de 27 de abril de 2016 (Parlamento Europeu, 2016).

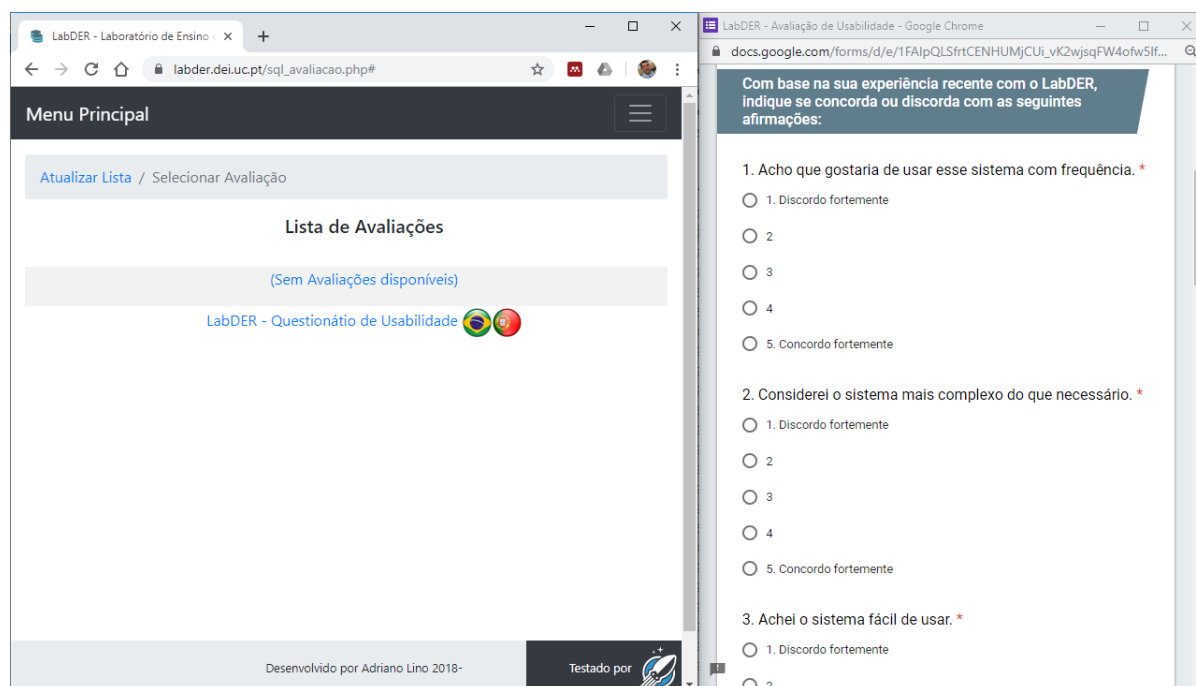


Figura 5-29. Implementação do formulário SUS no Google Forms e LabDER

A aplicação do questionário seguiu a seguinte rotina: (a) explicação da necessidade da aplicação do questionário; (b) explicação sobre a lei de proteção de dados; (c) explicação de que o questionário não está relacionado com os desempenhos escolares; (d) aplicação do questionário; (e) tabulação dos dados; e (f) geração de relatórios e gráficos.

5.5.4 Execução e Resultado

O teste foi executado conforme as agendas estabelecidas pelos professores e monitorizado pela ferramenta Google Analytics, ver exemplo de monitorização da universidade FAPAN do Brasil na (Figura 5-30).

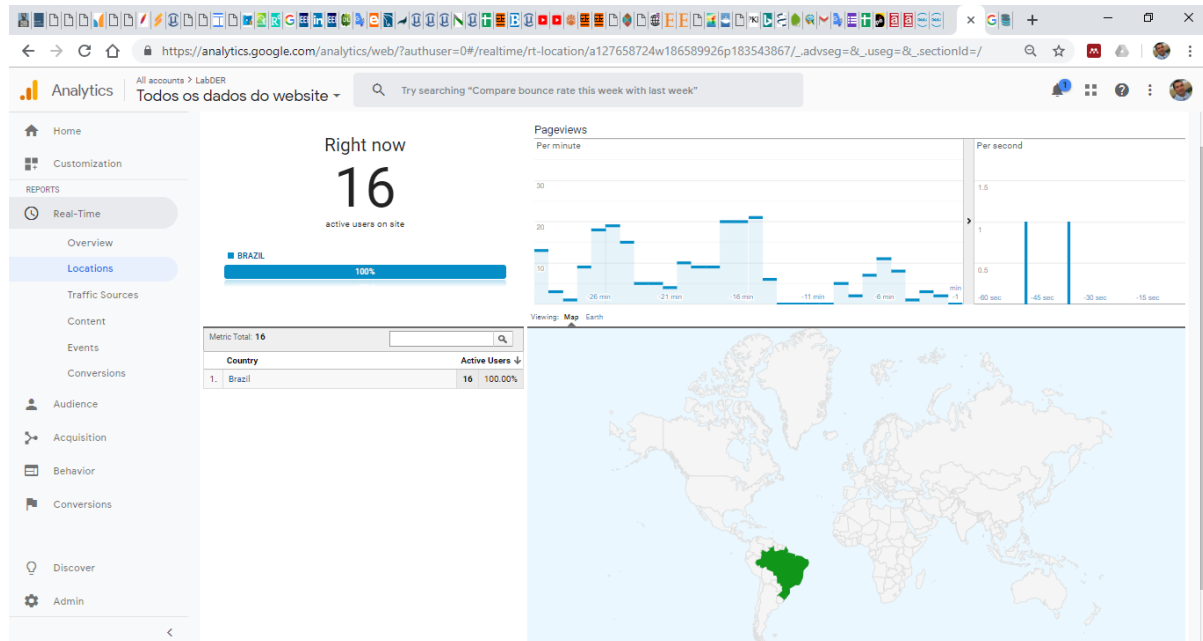


Figura 5-30. Acompanhamento da avaliação FAPAN.

Os dados foram recolhidos e tabulados conforme planeado. Ao todo, o LabDER foi avaliado por 64 utilizadores, e possui uma média de 72,97 na escala do SUS. A Figura 5-31 ilustra o resultado da avaliação do LabDER dentro da escala aceitável no SUS, especificamente dentro da escala “Bom”, conforme descrito por (Bangor et al., 2009).

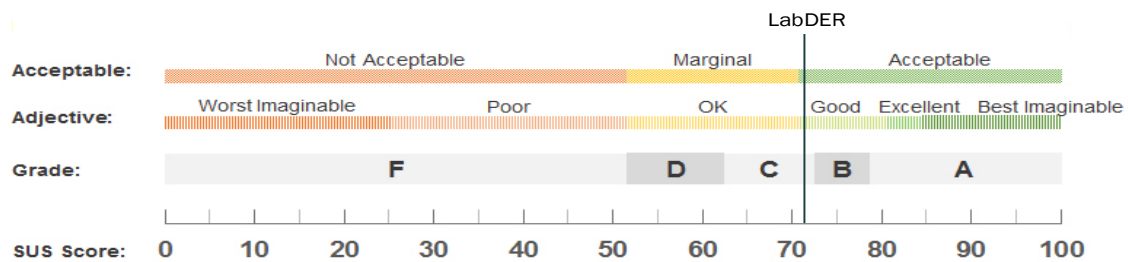


Figura 5-31. Classificação de usabilidade do LabDER (72,97) na escala de aceitabilidade (Bangor et al., 2009). As pontuações abaixo de 50 são consideradas inaceitáveis.

A frequência da avaliação pode ser vista na Figura 5-32, onde cerca de 13 utilizadores avaliaram o sistema na escala de aceitabilidade do “Melhor Imaginável”, e 25 utilizadores concordam que o sistema é “Excelente” dentro da aceitabilidade, enquanto 21 utilizadores

avaliaram o “Bom”. Apenas 1 utilizador avaliou o sistema quanto à aceitabilidade “Pior Imaginável”. Cerca de 98% dos utilizadores classificaram o sistema como aceite.

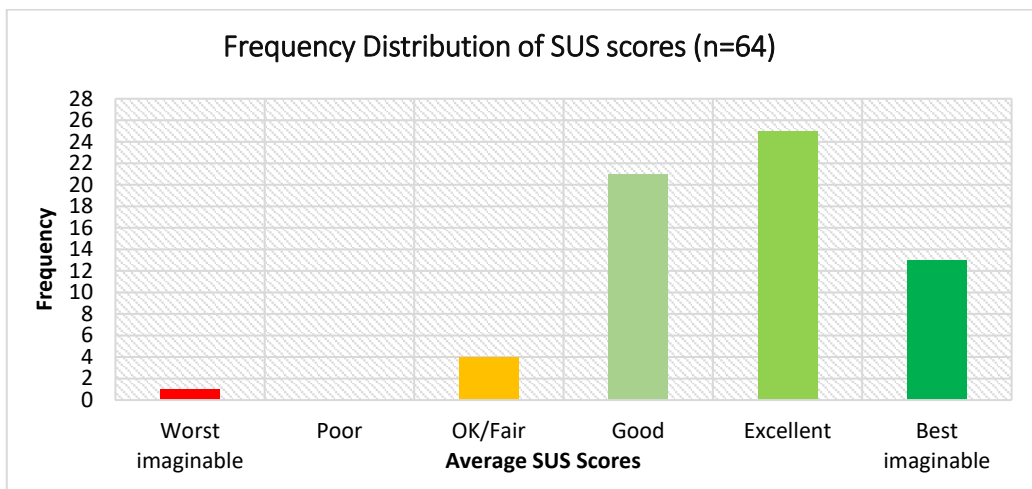


Figura 5-32. Distribuição de frequência de usabilidade do LabDER para classificação de adjetivos do SUS.

Com base na Figura 5-33, observa-se que 95% dos utilizadores concordam que o sistema tem uma classificação de 96,75 pontos, ou uma classificação de aceitabilidade da “Melhor Imaginável”. Cerca de 75% dos utilizadores consideram que o sistema tem uma classificação de 83,12 pontos, ou seja, concordam que o sistema tem uma classificação de aceitabilidade “Excelente”. Enquanto 50% dos utilizadores concordam que o sistema tem uma classificação de 75 pontos, ou seja, que o sistema tem a classificação “Bom” na escala de aceitabilidade do SUS. Por outro lado, cerca de 10% dos utilizadores concordam que o sistema tem uma classificação de 52,5 pontos, ou uma classificação “Ok” de aceitabilidade.

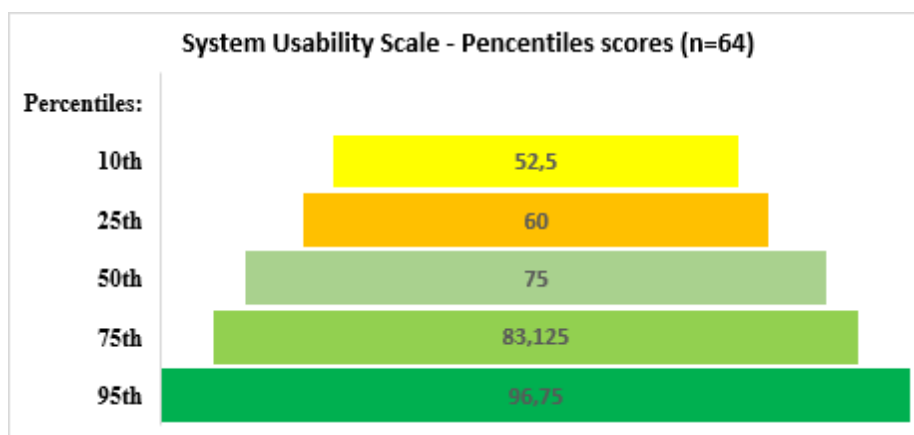


Figura 5-33. Classificações de usabilidade do LabDER em percentil, com 64 utilizadores.

Os testes de software foram aplicados com rigor para tentar demonstrar as evidências da satisfação do ponto de vista dos utilizadores (alunos e professores) sob a solução desenvolvida

que visa atender não só os requisitos dos utilizadores, mas também solucionar as limitações encontradas nas ferramentas anteriores. De entre as limitações encontradas tínhamos:

- A resposta restritiva: limitação a apenas uma resposta correta, em que os alunos são forçados a desenvolver um diagrama igual ao registado, porque o ambiente apenas aceita uma solução correta. Esta solução implementa uma abordagem de respostas múltiplas de computorização equivalente.
- Os destaques visuais: limita a reflexão dos alunos fornecendo a resposta esperada com destaques visuais nas palavras da descrição de uma questão. Esta solução implementa, na interface de execução de exercícios, três modalidades básicas de exercícios e suas combinações, que são do tipo de exercícios de transcrição, cenário e normalização. Assim, o professor combina os tipos de exercícios para criar uma questão, podendo ser mais simples ou difíceis de interpretar. No entanto, a interface nunca utiliza o registo do professor para indicar a resposta esperada, como ocorria nas abordagens anteriores.
- A ambiguidade: limitação de *feedback* oferecido pelas ferramentas para casos de diagramas equivalentes, em que os alunos podem obter um *feedback* correto por um motivo errado. A abordagem de avaliação automática desenvolvida neste projeto soluciona essa questão com equivalência entre gramáticas.
- O *feedback* estático: o sistema especialista reage ao conjunto de regras definidas estaticamente. Talvez, um conjunto de regras não tenha sido pensado e o sistema possa emitir um *feedback* errado. A abordagem trata o *feedback* na forma de compiladores, neste caso trata a resposta do aluno como uma língua, que é especificada pela notação *crow's foot* e, portanto, o tratamento de erros verifica se os conceitos de diagramas ER são violados ou não em relação à sua sintaxe e semântica.

Um conjunto de etapas bem estruturadas foram seguidas para tentar solucionar as limitações encontradas nas ferramentas de desenho e avaliação de DER, conforme a revisão da literatura. A nova abordagem de avaliação automática do diagrama de entidade e relacionamento foi proposta e implementada como produto de software, que é o LabDER. O resultado do produto foi aplicado e testado no contexto de sala de aula por professores e alunos de três países distintos. No final dos testes de software, há evidências suficientes para atestar que o LabDER é capaz de executar com precisão a recolha e tratamento de dados e pode ser usado por alunos e professores no processo de ensino e aprendizagem.

6

Implantação e Operação

6.1 Considerações iniciais

Este capítulo apresenta o projeto de implantação do LabDER, descrevendo quais os procedimentos adotados para aplicar o sistema em sala de aula e recolher os dados necessários para o desenvolvimento do modelo de avaliação automática.

Deste modo, são apresentadas as universidades convidadas, o plano de atividades adotado, e os materiais e métodos utilizados para garantir a aplicação do ensaio.

Também é descrito como os dados foram tratados a partir do método CRISP-DM para descobrir e validar o modelo de avaliação automática do DER.

6.2 Preparação do ambiente

Uma vez que o ambiente LabDER foi desenvolvido e testado, procurou-se estabelecer contactos com professores de BD para a aplicação prática do sistema, em sala de aula. Para isso, foi agendada uma vídeo conferência com os professores que estavam interessados no projeto. Nessa apresentação, foram estabelecidos os objetivos do projeto, o cronograma do projeto, o cronograma do curso baseado no plano de ensino do professor com um conjunto de atividades genéricas que seguiam, o desenvolvimento metodológico, o material didático envolvido e os requisitos informáticos necessários. Os itens da apresentação são descritos em detalhe no (Anexo E: Plano de atividades para uso do LabDER).

A Tabela 6-1 descreve o cronograma geral de atividades do projeto. É composto por quatro fases, são elas:

1. **Preparação do ambiente:** prepara o ambiente de trabalho para a utilização pelos participantes, período estimado de duas semanas. Deve iniciar-se antes do início das atividades académicas planeadas pelo professor. A documentação envolvida,

nesta fase, inclui uma lista com o nome completo e e-mail dos alunos que vão participar no projeto, as listas de exercícios com as questões e respostas, o plano de ensino do professor, indicando o período de duração das atividades com as datas de início e término, e a identificação completa do professor (instituição de ensino, nome completo do professor, nome do curso, nome da disciplina e identificação da turma).

2. **Execução das atividades:** é a fase de uso do ambiente de trabalho, em sala de aula, o momento para os alunos realizarem o conjunto de atividades previstas no plano de ensino do professor e implementadas no LabDER.
3. **Avaliação do ambiente de trabalho:** é a fase em que os participantes são convidados a avaliar o sistema. Essa atividade deve ser incluída após a última atividade prevista, no plano de ensino do professor, e antes do término das atividades acadêmicas.
4. **Encerramento do curso:** é a fase em que o investigador apresenta os resultados obtidos aos alunos e professores.

Tabela 6-1. Cronograma genérico de atividades do projeto.

#	Fase	Descrição	Período
1	Preparação do ambiente de trabalho	Registo dos utilizadores e questões. Documentos necessários. Plano de ensino do professor, lista(s) de exercícios com respostas, lista nominal dos alunos com e-mail.	2 Semana antes do início das atividades acadêmicas.
2	Execução das atividades	Período de interação dos participantes com o LabDER - realização dos exercícios/avaliações Atividade 01 - Tarefa (n questões) Atividade n - Tarefa (n questões) Atividade final - Avaliação (n questões)	Período reservado pelo professor.
3	Avaliação do ambiente de trabalho	Período de avaliação do ambiente pelos participantes. Responder ao questionário, pelo sistema, no dia determinado pelo professor.	1 Dia
	Agenda de Finalização	Encerramento do curso	1 Dia

Um volume expressivo de dados foi trocado entre os professores e o investigador, nessas quatro fases do projeto, em especial a primeira fase que usa uma série de documentos sensíveis, com os dados dos participantes e material do professor. Foi utilizado o Dropbox como ferramenta de controle de documentação (ver Figura 6-1). Basicamente os arquivos de modelo do sistema ficam na pasta “material do LabDER”, enquanto os arquivos do professor ficam na pasta “material do Professor”. São exemplos de informações trocadas entre o professor e o

investigador, plano de ensino do professor, lista de atividades e questões, material didático do curso, manual do sistema, entre outros documentos (ver Tabela 6-1).

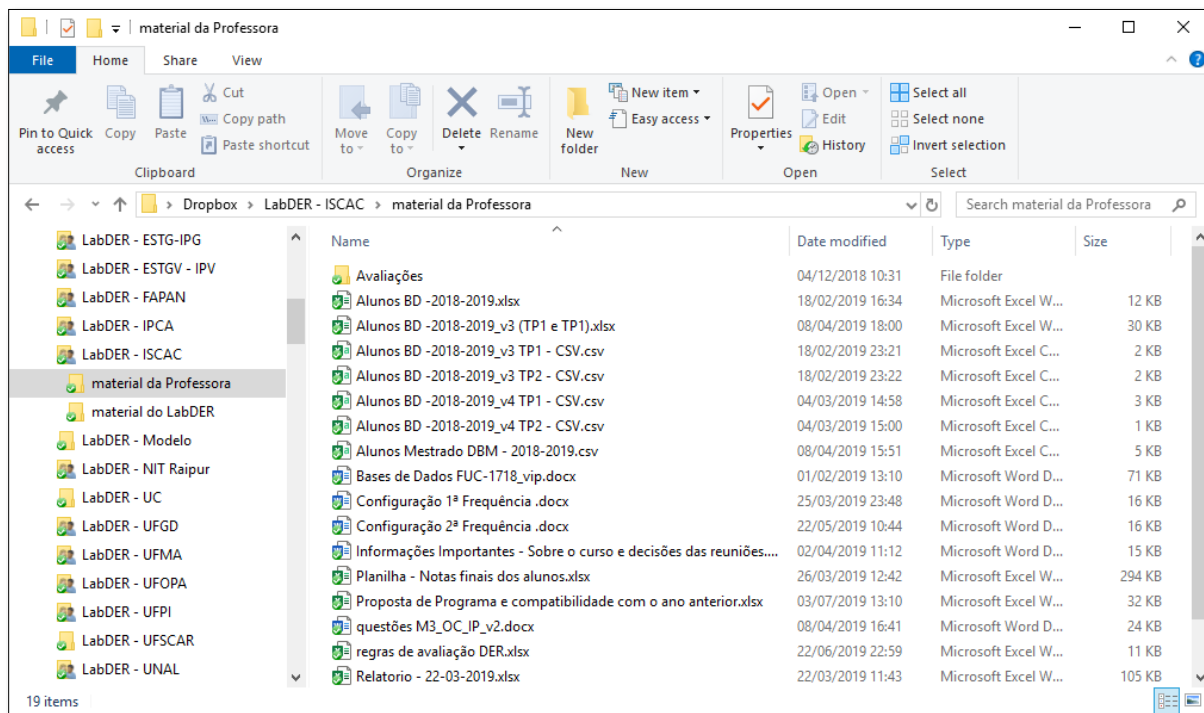


Figura 6-1. Controle dos dados trocados entre professores e investigador via DropBox.

O LabDER foi preparado para 14 universidades, sendo quatro em Portugal, com visitas e formações presenciais, e o restante à distância, sete no Brasil, uma na Índia e uma na Colômbia. Entretanto, algumas dessas universidades não iniciaram o projeto após a preparação do ambiente de trabalho, e outras limitaram-se ao módulo de SQL. Portanto, não contribuíram, efetivamente, na recolha de dados de DER. Ao todo, seis universidades contribuíram efetivamente na recolha de dados de DER (ver Tabela 6-2).

Tabela 6-2. Universidades que efetivamente utilizaram o módulo de DER.

Universidade	País	Link
Faculdade Pan Amazônica (FAPAN)	Brasil	http://www.suafaculdade.com.br/FAPAN/instituto/index.asp
Universidade Federal de Grande Dourados (UFGD)	Brasil	https://www.ufgd.edu.br/
Universidade Pública da Colômbia (UNAL)	Colômbia	http://www.manizales.unal.edu.co/
National Institute of Technology, Raipur (NIT)	Índia	http://www.nitr.ac.in/
Coimbra Business School (ISCAC)	Portugal	http://www.iscac.pt
Instituto Politécnico do Cávado e do Ave (IPCA)	Portugal	https://ipca.pt/

O controle de preparação e execução das atividades do LabDER era feito através do software Project (ver Figura 6-2), LabDER e Google Analytics. Desta forma, sabíamos as datas de avaliação formal, e os eventos principais que exigiriam suporte para garantir que o sistema rodasse em perfeitas condições, principalmente nos dias de exames formais.

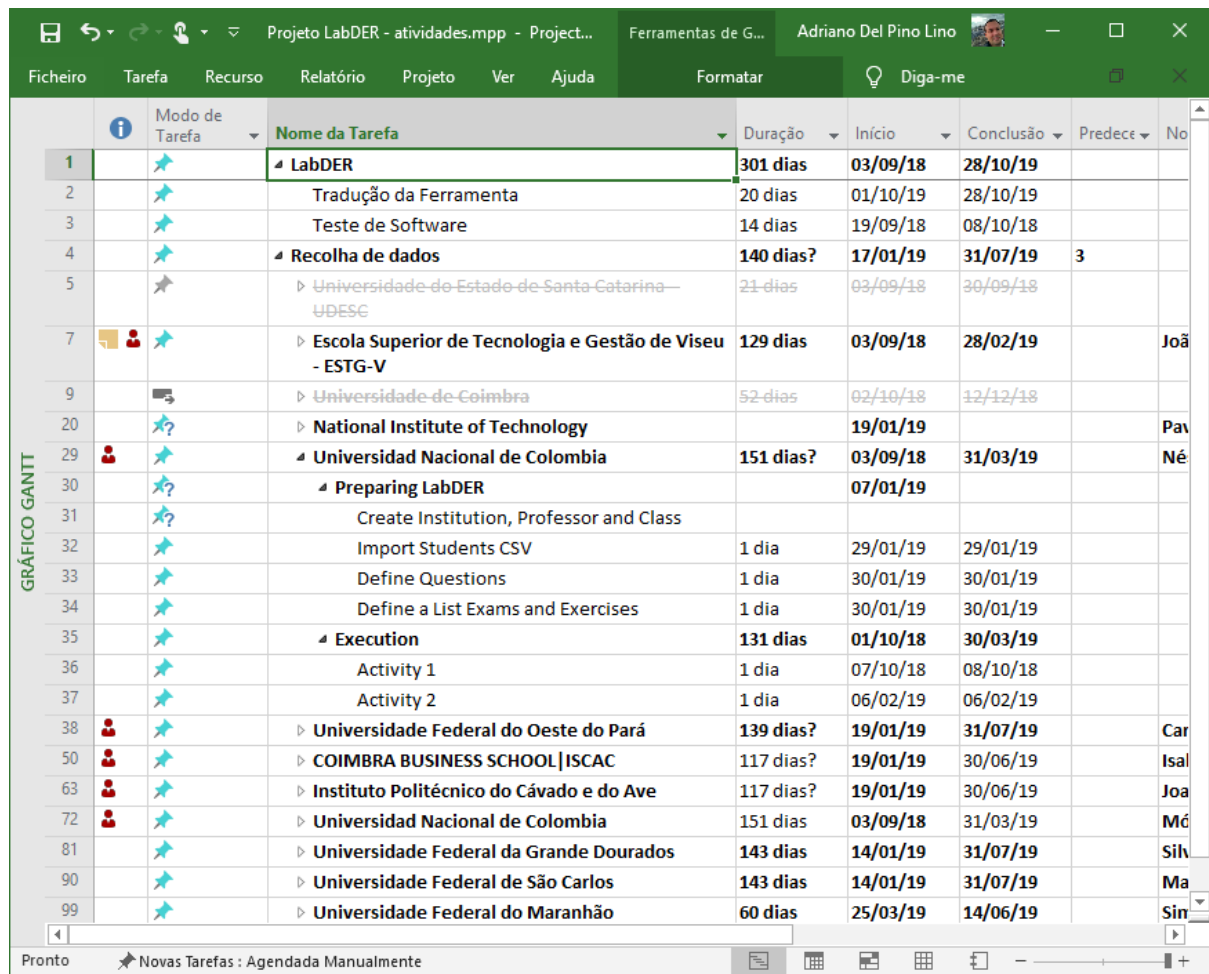


Figura 6-2. Cronograma de atividades de preparação e execução do LabDER no Project.

Na fase de preparação, uma sugestão personalizada da utilização da plataforma era criada a partir da análise do plano de ensino dos professores. A sugestão incluía todos os elementos do plano de ensino original do professor adaptados ao LabDER, como por exemplo, objetivos do curso, conteúdo programático e todo o conjunto de atividades planejadas pelo professor. Fundamentalmente, a sugestão é um mapeamento de cada um dos componentes do plano de ensino usando as funcionalidades do LabDER, começando pelo livro digital e depois pelas questões existentes (ver Figura 6-3). Um exemplo de mapa pode ser visto na Figura 6-3, que foi criado a partir do plano de ensino de um dos professores que usaram o LabDER em Portugal.

	A	B	C	D	E
1	Programa 15/03/2018			Material Existente LabDER	
2	1. Introdução às Bases de Dados			1 Módulo I - Introdução a Linguagem SQL	4
3	1.1 Bases de Dados: conceito e evolução			1.1 Histórico	4
4	1.2 SQL			1.2 Um banco de dados relacional	4
5	2. Modelos de Bases de Dados			1.2.1 Tabela, Linha e coluna.	5
6	2.1. Dados, Informação, Conhecimento			1.3 Esquemas de BD: modelagem E-R vs. SQL	6
7	2.2 Bases de Dados			1.4 Restrições de integridade	7
8	2.3. Sistemas Gestores de Bases de Dados			1.5 Usos do SQL: numa console cliente vs. embutida numa aplica	9
9	2.4. Modelos de Bases de Dados			1.6 Tipos de Instruções SQL	11
10	3. Desenho de Bases de Dados Relacionais			1.6.1 Instruções DDL	11
11	3.1. Ciclo de vida de uma Base de Dados			1.6.2 Instruções DCL	11
12	3.2. Diagrama Entidade - Associação			1.6.3 Instruções DML	11
13	4. Normalização			1.7 Elementos da sintaxe SQL	12
14	4.1 Dependências funcionais			1.8 Consultas em álgebra relacional	14
15	4.2 Formas normais			1.8.1 Seleção e Projeção	15
16	5. Seleção em Bases de Dados			1.8.2 Junção e Produto Cartesiano	16
17	SELECT, WHERE, operadores lógicos, operadores relacionais, BETWEEN, IN, IS LIKE			1.8.3 União, Intersecção e Diferença	16
18	6. Ordenação em Bases de Dados			1.8.4 Divisão	17
19	Uma coluna, várias colunas. NULL, DISTINCT e ALL			1.9 Questões	18
20	7. Junção de tabelas			2 Módulo II - SELECT (básico)	20
21	7.1 Produto Cartesiano de tabelas			2.1 Cláusulas FROM e WHERE	22
22	7.2 INNER JOIN, OUTER JOIN			2.1.1 Cláusula DISTINCT	23
23	7.3 Union, INTERSECT e MINUS			2.1.2 Cláusula WHERE	24
24	8. Funções de Agregação			2.2 Cláusula ORDER BY	24
25	COUNT, MIN, MAX, SUM e AVG			2.3 Tipos de operadores	25
26	9. Agrupar Informação			2.3.1 Expressões	26
27	GROUP BY, HAVING, WHERE, NULL			2.3.2 Operadores comparação: IN, IS NULL, BETWEEN	26
28	10. Subqueries			2.3.3 Comparações com LIKE, curingas (string)	27
29	11. Manipulação de Dados			2.3.4 Expressões com AND OR NOT e com parênteses	28
30	INSERT, UPDATE e DELETE			2.4 Exercícios	29
31	12. Criação e Manipulação de Tabelas			3 Módulo III - Manipulação de Tabelas	30
32	CREATE DATABASE, DROP DATABASE			3.1 Create Database	30
33	CREATE TABLE, ALTER TABLE, DROP TABLE			3.2 Create/drop table	31
34	13. VIEWS e ÍNDICES			3.2.1 Como duplicar uma tabela.	33
35				3.3 Insert/update/delete	34
36	14. Transações			3.4 Create view	35
37	COMMIT e ROLLBACK			3.5 BD emp, depto, dependente	35
38	CHECKPOINT			3.6 Transações	38
39	15. Segurança em Bases de Dados			3.7 Create index	38
40	16. Projeto de Bases de Dados			3.8 Restrições de FOREIGN KEY no MySQL	40
41	17. Data Warehouses			4 Módulo IV - GROUP BY	43
42					

Figura 6-3. Exemplo de mapeamento do conteúdo programático do professor e o material do LabDER.

A sugestão incluía um quadro de compatibilidade, que resume cada um dos itens que foram mapeados. O quadro apresenta em verde os conteúdos alinhados com o material do LabDER, e em vermelho alguma limitação do material didático, conforme na Tabela 6-3.

Tabela 6-3. Quadro com resumo da compatibilidade dos programas do professor e LabDER.

Programa Original do Professor	Programa do LabDER	Comentário
1. Introdução às Bases De Dados	Módulo I - Introdução a Linguagem SQL	Tudo igual
2. Modelos de Bases de Dados	Conteúdo Básico	Apresentamos no capítulo 1. Existem diversas questões que sobre os conceitos de SGBD, dados, chave primária, etc.
3. Desenho de Bases de Dados Relacionais	Módulo VI - Conceitos de diagrama de entidade e relacionamento	O LabDER apresenta os conceitos sobre o modelo conceptual implementados no modelo físico. A normalização faz parte do projeto do DER, mas ainda não temos o material didático. Aqui é esperada a utilização do LabDER nas listas de exercícios ou avaliações. Normalmente é exigido ao aluno a 3ª forma normal das questões existentes.
4. Normalização	Conteúdo Básico	
5. Seleção em Bases de Dados		
6. Ordenação em Bases de Dados	Módulo II - SELECT (básico)	
7. Junção de Tabelas	Módulo V - Subconsultas	
8. Funções de Agregação	Módulo IV - GROUP BY	
9. Agrupar Informação		Tudo igual
10. Subconsultas	Módulo V - Subconsultas	
11. Manipulação de Dados		
12. Criação e Manipulação de Tabelas	Módulo III - Manipulação de Tabelas	
13. Visão e Índices		
14. Transações	3.6 Transações (Apenas Sintaxe)	No LabDER tem apenas a sintaxe. Normalmente usamos os slides e livros para abordar esses conceitos e depois mostramos na prática no LabDER com as sintaxes.
	31 - Criar Bases de Dados (Sintaxe Segurança Grants)	No LabDER existe apenas uma pequena secção sobre a sintaxe 'Grant user' para falar sobre as possibilidades de segurança. Basicamente, explicamos a segurança implementada pelo SGBD com os 'donos' das tabelas, base de dados, e outros níveis que o SGBD implementa através de sintaxe Grant.
15. Segurança em Bases de Dados		
16. Projeto de Bases de Dados	Disponível para download (material de apoio)	

Em seguida, um conjunto de atividades e critérios de avaliação foi sugerido ao professor, com base no seu plano de ensino. Essas atividades estavam alinhadas com os objetivos e competências a serem adquiridas pelo aluno, que também estavam descritas no plano de ensino do professor. Por fim, o professor aperfeiçoou essas atividades, e decidiu exatamente a carga de trabalho que foi executada com os alunos. Então, a primeira fase de preparação do LabDER foi tida como concluída, uma vez que todas as atividades foram identificadas e inseridas na plataforma. A seguir, a Tabela 6-4 apresenta o resumo de atividades definidas pelo professor

que foram criadas no LabDER e executadas por alunos e professores durante um semestre letivo.

Tabela 6-4. Resumo das atividades a serem executadas no LabDER.

Metodologia de avaliação			
Atividades obrigatórias	Conteúdo LabDER	Comentário	Data Prevista
Atividade 1: Total de Questões 63 Módulo 1 - De texto livre - Qtd. Prob: 5 Módulo 1 - Alternativas fáceis - Qtd. Prob: 8 Módulo 1 - Alternativas difíceis... - Qtd. Prob: 3 Módulo 2 - De texto livre - Qtd. Prob: 3 Módulo 2 - Alternativas fáceis - Qtd. Prob: 2 Módulo 2 - Alternativas difíceis... - Qtd. Prob: 4 Módulo 2 - SQL - Qtd. Prob: 11 Módulo 4 - Alternativas - Qtd. Prob: 7 Módulo 4 - De texto livre - Qtd. Prob: 11 Módulo 4 - SQL - Qtd. Prob: 9	Módulo I - Introdução Módulo II - SELECT (básico) Módulo IV - GROUP BY	Nesse cenário, o projeto de BD é trocado por frequências.	18/02 a 25/03
Atividade 2: Total de 45 Questões Módulo 5 - De texto livre - Qtd. Prob: 6 Módulo 5 - Alternativas fáceis - Qtd. Prob: 7 Módulo 5 - Alternativas difíceis... - Qtd. Prob: 7 Módulo 5 - SQL - Qtd. Prob: 25	Módulo V - Subconsultas	Esse módulo inclui a complexidade de todos os módulos anteriores.	27/03 a 15/04
Atividade 3: Total de Questões 60	Módulo VI - Diagramas de entidade e relacionamento	Esse módulo possui diversas questões de DER, mas não tem questões de escolha múltipla. Neste caso, seria importante registrar as questões que acharem interessantes.	17/04 a 10/06
Frequência 1	Módulos 1,2 e 4	Todos os conceitos de SQL e introdutórios	26/mar
Frequência 2	Módulo 5	Subconsultas	16/abr
Frequência 3	Módulo 6	Avalia os conceitos sobre DER	11/jun

A classificação final é obtida aplicando a seguinte fórmula, cujo resultado é arredondado às unidades: $CF = (Ativ1 * 10\% + Ativ2 * 10\% + Ativ3 * 10\% + Freq1 * 35\% + Freq2 * 15\% + Freq3 * 20\%)$

Comentário: Nesse cenário, os exames representam 70% das atividades, e os exercícios representam os 30%

A quantidade de frequências e exames podem variar e pesos variam conforme tópico de interesse.

O conjunto de atividades descritos na Tabela 6-4 representa toda a carga de trabalho executada por dois professores numa das instituições de ensino em Portugal. Essas atividades resultam da aplicação do plano genérico de recolha de dados, que é adaptável ao plano de ensino do professor. Esse plano genérico foi então aplicado a cada professor, que aceitou participar no projeto. No final, ele decidia a carga de trabalho que gostaria de executar com os alunos utilizando o LabDER.

A segunda fase do projeto, execução do plano de atividades, foi realizada pelos professores e alunos recorrendo ao LabDER, que era justamente a execução do conjunto de

atividades, previamente definidas na primeira fase. Em alguns casos, a execução teve a participação do investigador, quer fosse à distância, como no Brasil, Índia e Colômbia, como presencialmente em Portugal. Listamos na Tabela 6-5 a quantidade de alunos e professores por instituições de ensino, que utilizaram a ferramenta e que contribuíram efetivamente na recolha de dados de DER. Ao todo foram 322 alunos e 8 professores.

Tabela 6-5. Quantidade de utilizadores por curso que tiveram atividades DER.

Universidade	Cursos	Alunos	Professores
UFGD	Engenharia de Computação (EG) Turma 1	32	
UFGD	Engenharia de Computação (EG) Turma 2	29	1
FAPAN	Ciência da Computação (ASI)	32	1
UNAL	Administração de Sistemas Informáticos (ASI) Turma 1	19	1
UNAL	Administração de Sistemas Informáticos (ASI) Turma 2	12	1
NIT Raipur	Tecnologia da Informação (TI)	44	1
IPCA	Licenciatura em Engenharia Eletrotécnica e de Computadores (EEC)	46	
IPCA	Curso Técnico Superior Profissional em Desenvolvimento Web e Multimédia (DWM)	44	1
IPCA	Licenciatura em Engenharia em Desenvolvimento de Jogos Digitais (EDJD)	41	
ISCAC	Licenciatura em Informática de Gestão (LIG) Turma 1	19	1
ISCAC	Licenciatura em Informática de Gestão (LIG) Turma 2	4	1
	Subtotal	322	8
	Total		330

Os cronogramas de execução de atividades de cada instituição de ensino são diferentes e, por isso, estabelecemos a data limite de recolha de dados para 11 de julho de 2019, que era a última avaliação formal registada no LabDER, de entre todas as universidades que participaram na recolha de dados de DER.

A execução das atividades ocorreu sem problemas. Todas as manutenções no sistema eram previamente agendadas e as mudanças de datas de cada atividade eram tratadas com normalidade pelo sistema, bem como a separação das atividades em subdivisões. No período de execução das atividades, evidenciamos uma triangulação entre Colômbia (alunos e professor auxiliar), Espanha (professor adjunto) e Portugal (investigador) para garantir a execução da primeira avaliação formal na UNAL, que ocorreu em 06 de fevereiro de 2019. A partir do log do LabDER e dos dados do Google Analytics eram identificados os utilizadores ativos na plataforma, com a identificação de dispositivo de hardware, sistema operacional utilizado, localização, velocidade de acesso, entre outras informações (ver Figura 6-4).

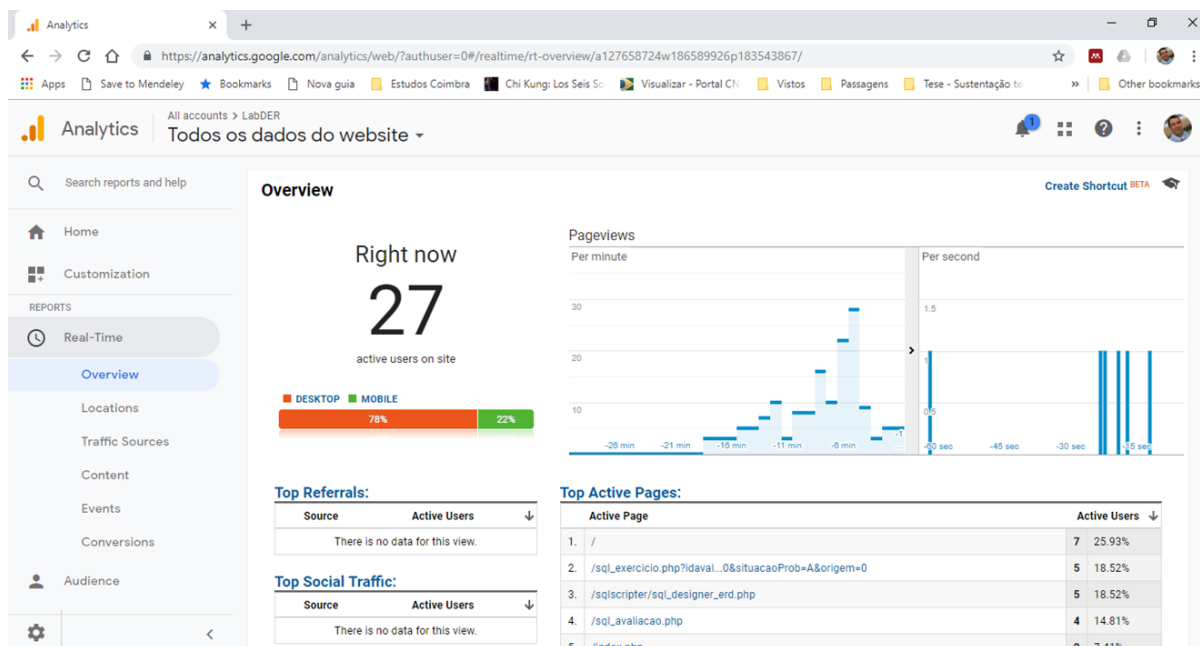


Figura 6-4. Utilizadores ativos na primeira avaliação formal da UNAL.

Uma lentidão foi reportada na primeira avaliação, e rapidamente identificado o motivo. Aqui foi diagnosticado que uma regra de segurança implementada no login do LabDER, a regra de bloqueio de acesso com 5 tentativas de erro, também bloqueava o endereço IP da instituição de ensino. A solução foi registrar os IPs das universidades, o que foi repetido para as demais instituições participantes deste projeto. Não foram encontradas dificuldades de execução, e nenhum outro problema foi relatado pelos professores ou alunos, e todo o processo de recolha seguiu, conforme o planeamento de atividades dos professores.

No final da execução de todas as atividades registadas pelos professores, daquelas universidades que tiveram atividades de DER, foram recolhidas mais de 40 mil respostas dos alunos, sendo 5.480 de diagramas ER, 2.908 de questões de texto livres, 3.185 questões de escolha múltipla e 13.781 questões de SQL, e 15.377 submissões de SQL, em modo treino (ver Figura 6-5).

Soma de Qtde	Rótulos de Coluna					
Rótulos de Linha	DER	SQL	SQL-Livre	Escolha Múltipla	Texto Livre	Total Geral
FAPAN	117	989		338	329	1 773
IPCA	600	6 409	391			7 400
ISCAC	4 089	6 114	14 314	2 472	2 539	29 528
NIT Raipur	188		3			191
UFGD	408		1	2	3	414
UNAL	78	269	668	346	37	1 398
Total Geral	5 480	13 781	15 377	3 158	2 908	40 704

Figura 6-5. Total de submissões no LabDER por tipo de questão.

A frequência das submissões dos alunos, por tipo de questões e universidade, pode ser vista na Figura 6-6. Os alunos do ISCAC executaram mais atividades e, conseqüentemente, submeteram mais respostas ao sistema.

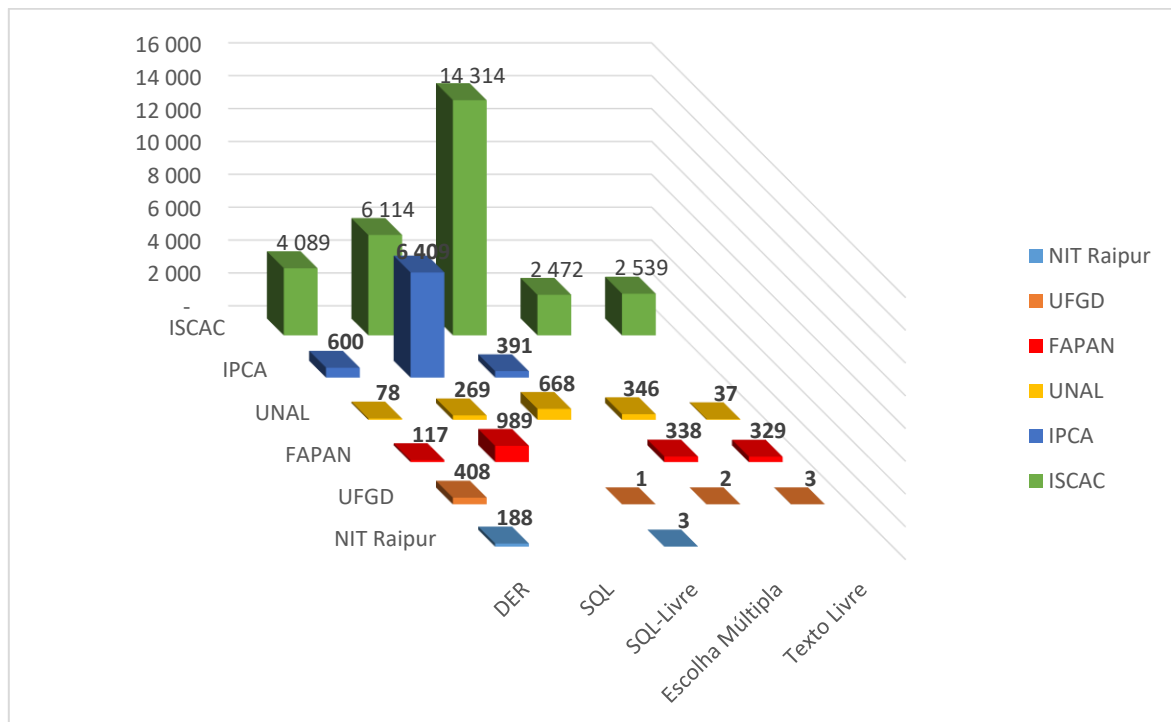


Figura 6-6. Gráfico da frequência de submissões por tipo de questão e universidade.

Todos os dados e todas as informações sobre o diagrama ER (respostas dos alunos, resposta do professor e avaliação do professor) recolhidos foram armazenados na BD do LabDER. Depois das classificações serem recolhidas, foi verificada a integridade dos dados. Dado que um conjunto completo de respostas foi obtido e todos os indivíduos tinham experiência suficiente na avaliação de construção de DER, trabalhámos com a premissa de que as avaliações dos professores eram fiáveis o suficiente para atender aos objetivos deste estudo.

A implantação do LabDER foi executada, em momentos distintos, em cada universidade e, por isso, iniciou-se o desenvolvimento do processo de conceção do modelo preditivo, logo após obtermos um conjunto de dados de DER. Este processo é descrito em formas de ciclos, usando uma metodologia específica, que representa o esforço dedicado para obter o melhor modelo a ser utilizado pelos participantes do projeto.

6.3 Processo de concepção do modelo

O processo de desenvolvimento do modelo de aprendizagem computacional escolhido neste projeto é baseado na metodologia *CRoss Industry Standard Process for Data Mining* (CRISP-DM), que tem sido largamente utilizada na literatura (Shafique & Qaiser, 2014).

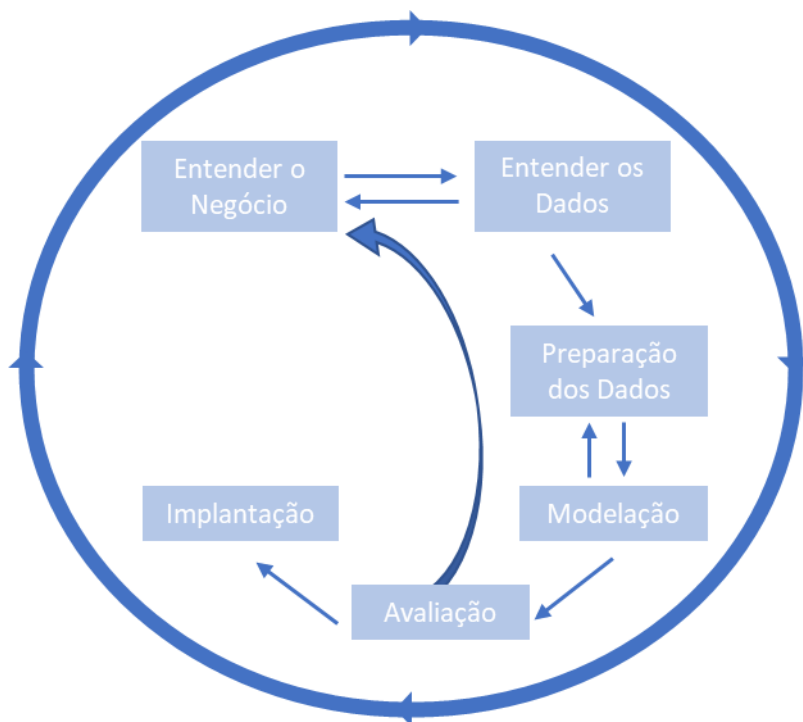


Figura 6-7. Modelo CRISP-DM (Chapman et al., 2000).

O método CRISP-DM (ver Figura 6-7) segue seis etapas: (1) entender o negócio, que visa desenvolver as metas do estudo; (2) entender os dados, que identifica e auxilia compreender as fontes de dados relevantes, (3) preparação dos dados, que limpa e transforma os dados relevantes, (4) modelação, que desenvolve modelos usando técnicas analíticas comparáveis, (5) avaliação, que procura avaliar os modelos uns contra os outros e contra os objetivos do estudo, e (6) implantação que mostra o resultado da implantação do modelo (Chapman et al., 2000).

O processo de concepção do modelo de avaliação de DER seguiu um guia “passo a passo” do método CRISP-DM, proposto por Chapman et al (2000), que sugere um conjunto de atividades com saídas esperadas para cada uma das etapas do método, que é resumidamente ilustrada na Figura 6-8.

Entender o Negócio	Entender os Dados	Preparação dos Dados	Modelação	Avaliação	Implantação
Determinar objetivos de negócio: Contexto; Objetivos de negócio; Determinar os critérios de sucesso do negócio; Avaliar situação: Inventário de Requisitos de Recursos; Pressupostos e restrições; Riscos e contingências; Terminologia; Custos e benefícios; Determinar metas de MD: Objetivos de MD; Critérios de sucesso para MD; Produzir plano de projeto: Plano de projeto; Avaliação inicial de ferramentas e técnicas;	Coletar dados iniciais: Coleta inicial de dados; Relatório; Descrever dados: Descrição de dados; Relatório; Explorar dados: Exploração de dados; Relatório; Verificar a qualidade de dados: Relatório de qualidade de dados;	Selecionar dados: Justificativa para inclusão/exclusão; Limpar dados: Relatório de limpeza de dados; Construir dados: Atributos derivados; Registros gerados; Integrar dados: Dados mesclados; Formatar dados: Dados reformatados; Conjunto de dados; Descrição do conjunto de dados;	Selecionar algoritmo de modelação: Algoritmo de modelação; Pressupostos dos algoritmos de modelação; Gerar projeto de teste: Projeto de teste; Construir o modelo: Parâmetros de configurações; Modelos; Descrições dos modelos; Avaliar modelo: Avaliação do modelo; Configurações de parâmetros revisadas;	Avaliar resultados: Avaliação dos resultados da MD com os critérios de sucesso nos negócios; Modelos aprovados; Processo de revisão: Revisão de processo; Determinar os próximos passos: Lista de ações possíveis; Decisão;	Planejar a implantação: Plano de preparação; Planejar Monitoração e Manutenção: Plano de monitoração e manutenção; Produzir relatório final: Relatório final; Apresentação final; Revisão do Projeto: Documentação de experiência;

Figura 6-8. Atividades do CRISP-DM, extraído de (Chapman et al., 2000).

Foram aplicados três ciclos experimentais que contemplaram as seis etapas do método CRISP-DM. A descrição das atividades e resultados obtidos em cada etapa estão descritos nas três secções seguintes.

6.3.1 Primeiro ciclo

O processo baseado em CRISP-DM oferece uma maneira sistemática e estruturada de conduzir estudos de mineração de dados e, assim, aumentar a probabilidade de obter resultados precisos e fiáveis. Neste primeiro ciclo, é dada uma atenção especial às etapas de entender o negócio, entender os dados e na preparação dos dados, na intenção de ser bem-sucedido na procura do modelo. Aproximadamente 80% do tempo total do projeto é normalmente gasto nessas três primeiras etapas (Chapman et al., 2000).

6.3.1.1 Entendimento do Negócio

Segundo Chapman et al. (2000), a fase de entendimento do negócio concentra-se em entender os objetivos e requisitos do projeto de uma perspectiva de negócios, convertendo esse conhecimento numa definição de problema de mineração de dados e num plano preliminar projetado para alcançar os objetivos (Chapman et al., 2000).

Contexto: O principal objetivo desta investigação é determinar um modelo de avaliação automática de DER que mais se aproxime das avaliações do professor. Este objetivo convertido na perspectiva de mineração de dados é encontrar o melhor modelo para prever a classificação do professor, durante o processo de avaliação do diagrama ER do aluno. Os dados para treinamento do modelo, resposta DER dos alunos e classificações dos professores, foram recolhidos em distintos momentos conforme descrito na seção 6.2 Para atingir a meta de mineração de dados, devem-se comparar diferentes algoritmos de aprendizagem computacional supervisionada e devem ser treinados diversos modelos sobre os dados

contendo um conjunto de atributos que descrevem as características dos diagramas ER. Posteriormente, os dados de desempenho desses modelos serão confrontados entre si para avaliar e definir o melhor modelo a ser implantado no LabDER.

Os objetivos de negócio de mineração de dados no contexto desta investigação são:

1. Prever a avaliação do DER do aluno, pelo seu histórico de respostas com informações do DER (número de tabelas, colunas, relacionamentos, etc.) e da avaliação do professor. Quando o aluno submete o DER ele deve receber a classificação de forma automática;
2. Encontrar o melhor algoritmo preditivo, para isso deve-se comparar os resultados dos desempenhos de diferentes algoritmos de aprendizagem computacional supervisionada.

O critério de sucesso depende do contexto de aplicação da aprendizagem computacional (Portugal, Alencar, & Cowan, 2018). Por exemplo, no diagnóstico médico são procuradas valores de AUC muito altas com 95% ou mais (Christodoulou et al., 2019). Na psicologia os valores da AUC de 70% e superiores seriam considerados fortes (Rice & Harris, 2005). Na educação os valores de *accuracy* variam entre 56% a 98% dependendo do algoritmo utilizado (Shahiri, Husain, & Rashid, 2015). Segundo Hosmer e Lemeshow (2005), em geral, os valores de AUC de 50% sugere nenhuma distinção (ou seja, capacidade de classificar o desempenho do aluno em aprovado ou reprovado), 70% a 80% é considerado aceitável, 80% a 90% é considerado excelente e mais de 90% é considerado excepcional (Hosmer & Lemeshow, 2005). Para esta investigação, considerando a qualidade mínima esperada de um modelo preditivo aplicado no contexto educacional, definiu-se os seguintes critérios de sucesso:

1. Os modelos devem ter desempenho mínimo de 90% de *accuracy* e *precision*;
2. O algoritmo mais eficiente é aquele que possui a maior média de desempenho de *precision*, *recall*, *F1-Score* e *accuracy*;
3. Os algoritmos devem ter uma taxa máxima de 5% de desvio padrão, nos desempenhos de *precision*, *recall*, *F1-Score* e *accuracy*;
4. No caso de vários algoritmos eficientes, o critério de desempate é aquele que possuir maior desempenho de *recall*. No contexto educacional, não são desejáveis sistemas com baixo *recall*, pois prefere-se um modelo que não seja injusto na classificação dos alunos que responderem corretamente.

Riscos e contingências:

1. Dados pobres, que possuem baixa qualidade para gerar modelos:
 - a. Considerar que as respostas dos professores estão incorretas ou ambíguas, identificar, corrigir ou eliminar os erros em conjunto com os professores;
 - b. Descobrir as limitações da abordagem e limitar o propósito da avaliação;
 - c. Considerar outras formas de medir a distância entre DER;
 - d. Considerar uma nova recolha de dados.
2. Os modelos fornecem classificadores com taxa menor que 80% de *Area Under the Curve* (AUC):
 - a. Considerar excluir, agrupar ou derivar as métricas;
 - b. Considerar o uso de técnicas de aprendizagem computacional profunda;
 - c. Restringir a investigação, e focar em mudanças de parâmetros nos algoritmos com melhores desempenhos.

O método CRISP-DM é aplicado repetitivamente em formas cíclicas, aperfeiçoando o modelo até satisfazer o objetivo de encontrar um modelo de avaliação automática de DER com no mínimo 90% de *precision*. As repetições são aplicadas a fim de investigar, mais a fundo, as limitações de cada modelo e encontrar alternativas para alcançar os critérios de sucesso. Neste caso, as repetições são uma busca sistemática para sustentar a hipótese de que as métricas de DER são suficientemente boas para gerar um modelo preditivo com 90% de *accuracy*.

6.3.1.2 Compreensão dos Dados

A fase de Compreensão dos Dados descreve, sob o ponto de vista do processo de aprendizagem computacional, a recolha de dados e prossegue com atividades para se familiarizar com eles. Independente do ciclo, esta fase é útil para identificar problemas de qualidade de dados e descobrir os primeiros *insights* ou detetar subconjuntos interessantes para formar hipóteses para informações (Chapman et al., 2000).

Os dados recolhidos estão na BD do LabDER, que contém todas as informações relacionadas com as atividades desenvolvidas por alunos e professores. O volume de dados selecionados, neste primeiro ciclo é de 1.313 registos. Os dados estão armazenados em tabelas relacionais e as principais tabelas envolvidas neste primeiro nível de compreensão dos dados são:

- Dados das respostas dos alunos: A tabela de respostas do LabDER contém todas as respostas dos alunos (DER, SQL, escolha múltipla, e de texto livre), num campo do tipo texto. Cada resposta tem a respetiva avaliação do professor, que é uma nota armazenada no campo do tipo numérico. Diversos identificadores permitem relacionar com outras tabelas da BD, como a tabela dos professores e questões.
- Dados das respostas do professor: A tabela de repostas contém todas as respostas registadas pelos professores.
- Dados das métricas: Em cada atributo da tabela de métrica são armazenadas as informações sumarizadas de métricas extraídas de cada resposta DER, seja do aluno ou professor. Todas os atributos são do tipo numérico. Tabelas auxiliares armazenam informações detalhadas de cada elemento que compõe uma resposta DER, que são úteis e podem ser usadas no processamento de dados.

Conforme Chapman (2000) “Os dados raramente são perfeitos. Na verdade, a maioria dos dados contém erros de codificação, valores ausentes ou outros tipos de inconsistências que dificultam a análise.”. O SGBD foi usado para eliminar a ocorrência desses problemas, pois o SGBD garante a integridade do dado. Por exemplo, não existem dados de texto em colunas do tipo numérico, ou valores vazios em atributos obrigatórios. No aspeto de qualidade do dado, não existem problemas de: dados ausentes, erros de dado, medidas de erro, inconsistência de codificação. A única exceção surge na incompatibilidade do tipo da avaliação do professor, que é numérico, que, neste caso, é transformado em categórico por uma expressão SQL.

Os dados brutos do DER do aluno e da resposta do professor podem ser transformados de duas formas: como uma junção ou como uma diferença. Neste caso, de acordo com a Figura 6-9, a linha 1 contém os dados referente à resposta do aluno e a linha 2 a resposta do professor.

1. Dados de entrada como junção: Nesse cenário, tem-se o dobro da quantidade de entradas a ser processado pelos algoritmos de aprendizagem computadorizada. Se tem 27 métricas para cada DER, então seriam cerca de 54 atributos. Alguns algoritmos têm limitações de quantidade de entradas, quanto maior for a entrada, maior será o tempo de processamento e talvez pior o desempenho.
2. A diferença das entradas: A diferença limita a quantidade de entradas para exatamente o mesmo número de métricas concebidas no sistema, que é de 27. A diferença entre essas informações faz lembrar o processamento cognitivo de

corrigir uma prova em papel. No qual, normalmente se confere qual é a diferença entre as respostas do aluno e professor, e então calcula-se a nota final. De forma análoga, a diferença bruta entre os registros 1 e 2 é de 1, pois são diferentes apenas na última coluna representada pela métrica de número de relacionamentos MN.

3. A saída esperada: É a avaliação do professor representada nos cinco rótulos linguísticos (1-Falhou 0% a 20%, 2-Fraco 21% a 49%, 3-Suficiente 50% a 69%, 4-Bom 70% a 89%, e 5-Excelente 90% a 100%). Esse dado bruto é armazenado como valor numérico na BD, e segue o devido tratamento de dados para representar as cinco classes de avaliação do professor.

	CD_Der	DS_ER	NR_Tbl	NR_Cln	NR_Rel	NR_RelUnr	NR_RelBin	NR_RelTrn	NR_RelIN	NR_RelMN	N
<input type="checkbox"/>	1 <.	3K	4	9	3	0	1	0	3	0	
<input type="checkbox"/>	2 <.	3K	4	9	3	0	1	0	3	1	
<input type="checkbox"/>	3 <.	3K	4	9	3	0	1	0	3	1	
<input type="checkbox"/>	4 <.	3K	3	14	2	0	1	0	0	0	
<input type="checkbox"/>	5 <.	3K	3	14	2	0	1	0	2	1	
<input type="checkbox"/>	6 <.	3K	3	14	2	0	1	0	0	0	
<input type="checkbox"/>	7 <.	3K	3	14	2	0	1	0	2	1	
<input type="checkbox"/>	8 <.	3K	3	14	2	0	1	0	2	1	
<input type="checkbox"/>	9 <.	3K	3	14	2	0	1	0	2	1	
<input type="checkbox"/>	10 <.	3K	3	14	2	0	1	0	2	1	
<input type="checkbox"/>	11 <.	3K	3	19	0	0	0	0	0	0	
<input type="checkbox"/>	12 <.	6K	6	43	6	0	1	1	5	2	
<input type="checkbox"/>	13 <.	4K	3	25	0	0	0	0	0	0	
<input type="checkbox"/>	14 <.	6K	6	42	6	0	1	1	3	1	
<input type="checkbox"/>	15 <.	6K	6	42	6	0	1	1	3	1	
<input type="checkbox"/>	16 <.	6K	6	42	6	0	1	1	4	1	
<input type="checkbox"/>	17 <.	6K	6	42	6	0	1	1	5	2	
<input type="checkbox"/>	18 <.	6K	6	42	6	0	1	1	5	2	
<input type="checkbox"/>	19 <.	2K	2	8	1	0	0	0	1	0	
<input type="checkbox"/>	20 <.	3K	2	8	1	0	0	0	1	0	
<input type="checkbox"/>	21 <.	3K	2	5	1	0	0	0	1	0	
<input type="checkbox"/>	22 <.	2K	2	5	1	0	0	0	1	0	
<input type="checkbox"/>	23 <.	3K	3	10	2	0	1	0	2	0	

Figura 6-9. Dados brutos da tabela de métricas.

A compreensão dos dados é fundamental para perceber as diferentes estratégias que podem ser aplicadas na concepção do modelo. A proposta inicial é aplicar a distância normalizada entre a resposta do aluno e a resposta do professor como entrada do sistema. Esta é uma prática desenvolvida na aplicação dos algoritmos de aprendizagem computacional, uma vez que “o método de normalização pode ser usado para obter melhor convergência e o novo dimensionamento pode fazer com que o algoritmo de aprendizagem computacional funcione melhor” (Wan, 2019).

6.3.1.3 Preparação dos Dados

Os dados são normalizados na escala [0,1] usando uma transformação de distância mínimo e máximo entre os atributos do DER do aluno e os atributos do DER do professor,

onde 0 significa que não há diferença entre os DER, e 1 significa que os DER são totalmente diferentes. Segundo Juszczak, Tax e Duin (2002), o método min.-max. diminui o erro nos dados de *outlier*, na maioria dos casos (Juszczak, Tax, & Duin, 2002). A fórmula é expressa como:

$$z_i = \frac{x^i - \min(x)}{\max(x) - \min(x)}$$

Uma das características deste estudo é que nem sempre as amostras de dados vão conter todos os conceitos necessários de DER para gerar um modelo de aprendizagem. Isto porque os professores priorizam determinados conceitos e preterem outros. Para detetar a importância de cada conceito, foi utilizado a seleção dos atributos baseada na análise de correlações e na frequência dos atributos.

Para detetar se um conceito de DER está presente na amostra de dados é usado a medida de frequência, e para medir a relação entre a avaliação do professor e os atributos de DER é usado a correlação. A medida de correlação é aplicada para decidir sobre a manutenção ou exclusão de um atributo em ML (Kuhn & Johnson, 2013; Kumar & Chong, 2018). A medida de correlação é também importante porque fornece informações que podem auxiliar na formação de novos atributos, por exemplo, usando combinação de atributos com baixa e forte correlação (Blum & Langley, 1997; Kumar & Chong, 2018).

A frequência em estatística (ou frequência absoluta) de um evento i é o número n_i de vezes que o evento ocorreu em um experimento ou estudo (Bluman, 2017; Illowsky & Dean, 2013). A frequência igual a zero indica que o atributo não foi utilizado no caso de estudo e, portanto, deve ser excluída na fase de modelação (Blum & Langley, 1997). Atributos com frequências abaixo de 100 são consideradas baixas, porque este valor é um número aproximado resultante da multiplicação da menor média da quantidade de respostas de DER por instituição vezes a quantidade de alunos da instituição. Para este experimento, considerou a menor média de respostas que é de 3,12 da instituição UNAL que têm 31 alunos, portanto, o limite de respostas é igual a 96,72. Assim, o valor 100 corresponde o menor valor aproximado de frequência de resposta. Os casos de baixa frequência devem ser examinados cuidadosamente pois podem representar uma questão muito específica elaborada pelo professor e respondida pelos alunos, ou simplesmente uma ausência de questões elaboradas pelo professor que abrangessem aqueles atributos de DER.

Em estatística, a correlação de *pearson* afere a relação linear entre duas variáveis quantitativas. É medida pela covariância de duas variáveis dividida pelo produto de seus

desvios-padrão, portanto, é essencialmente uma medida normalizada da covariância, de modo que o resultado sempre fica com um valor entre -1 e 1. Um valor de 1 implica que uma equação linear descreve a relação entre X e Y perfeitamente, com todos os pontos de dados situados em uma linha para a qual Y aumenta à medida que X aumenta. Um valor de -1 implica que todos os pontos de dados estão em uma linha para a qual Y diminui à medida que X aumenta (Bluman, 2017).

A correlação aplicada na amostra de dados de cada ciclo do CRISP-DM evidencia a força da relação entre os atributos de DER e a avaliação do professor. Considera-se a exclusão do atributo de DER quando não há correlação entre os atributos de DER e o atributo da avaliação do professor. A exclusão de um atributo de DER também é indicada quando a amostra de dados apresenta multicolinearidade (Cai, Luo, Wang, & Yang, 2018; Hirao, Okumura, Yasuda, & Isozaki, 2007; Kuhn & Johnson, 2013), ou seja, quando há uma relação linear muito aproximada entre dois ou mais atributos.

A seleção de atributos tem sido um tópico de pesquisa de há décadas, e os métodos de seleção podem ser baseados em estatísticas (Gibert, Valveny, & Bunke, 2012; Khoshgoftaar, Dittman, Wald, & Fazelpour, 2012; Lane, Xue, Liu, & Zhang, 2014; H. Li, Li, Wu, & Sun, 2014; Vasconcelos & Vasconcelos, 2009), teoria da informação (Bonev, 2010; Burnham & Anderson, 1998; Fleuret, 2004; Peng, Long, & Ding, 2005; Shen & Bai, 2006; H. H. Yang & Moody, 1999), variedade (Jie, Zhang, Cheng, & Shen, 2013; B. Li, Zheng, & Huang, 2008; Xu, King, Lyu, & Jin, 2010) e conjunto aproximado (Y. Chen, Miao, & Wang, 2010; Derrac, Cornelis, García, & Herrera, 2012; Shu & Shen, 2014; Swiniarski & Skowron, 2003; Wang, Guo, & Wang, 2010). O método de seleção baseado em estatística é categorizado pelo método de avaliação que pode ser derivado de correlação, distância euclidiana, consistência, dependência e medida de informação (Cai et al., 2018). Segundo Cai et al. (2018), o estado da arte da seleção de atributos online inclui o algoritmo SAOLA (Yu, Wu, Ding, & Pei, 2016), que faz a análise de redundância por correlação entre os atributos (Cai et al., 2018). Por mais que a seleção de atributos forneça uma maneira eficaz de eliminar os atributos redundantes ou irrelevantes, este método não é adequado para esta pesquisa. Ao invés, o método base de avaliação baseado em correlação é aplicado.

Nesta pesquisa, a seleção de atributos não é adequada porque poderia excluir um atributo importante do DER, ou associar atributos que não sejam conceptualmente aceitáveis em BD. Por exemplo, ao aplicar o algoritmo SelectionKBest², foram excluídos os atributos de

² https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

cardinalidades. A cardinalidade é um dos princípios fundamentais de BD relacionais. Nela são definidos os graus de relação entre duas tabelas (por exemplo, 1:N, N:N, 1:1) (Coronel & Morris, 2019). Por conseguinte, um modelo preditivo baseado na seleção de atributos implica num modelo de avaliação limitado. Por esta razão optou-se pelas medidas de correlação e frequência na fase de preparação dos dados. Pretende-se, assim, investigar as limitações de cada amostra de dados utilizada nos ciclos experimentais do CRISP-DM, e encontrar alternativas com a finalidade de manter no modelo preditivo o maior número de atributos de DER.

As correlações dos atributos de entrada são ilustradas na Figura 6-10, isto é, a correlação entre os atributos de entrada (distância dos atributos do DER do aluno e do professor) e a saída (o atributo de avaliação do professor). As correlações das variáveis são descritas no Anexo F: Métricas - Análises de Correlação - 1º Ciclo.

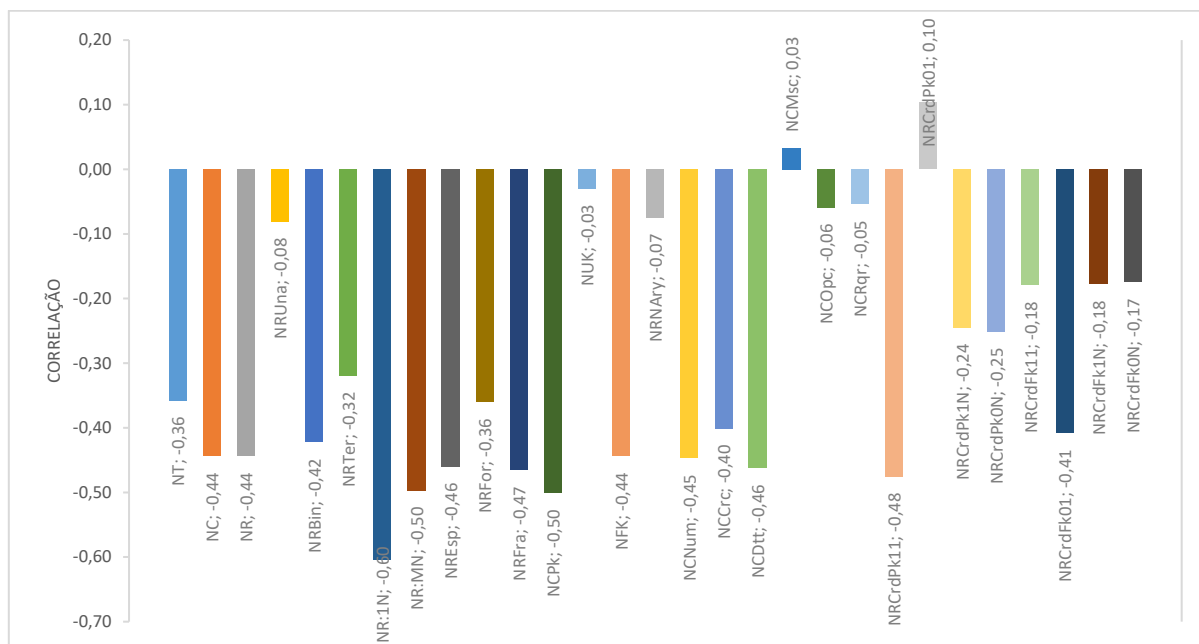


Figura 6-10. Importância das variáveis de entrada no 1º ciclo.

Os atributos NUK (número de colunas chaves únicas) e NCMsc (número de colunas miscelâneas) têm correlações muito fracas com a avaliação do professor, próximo de 0 (ver Figura 6-10). Isso ocorre porque os professores não criaram um conjunto de atividades que avaliassem essas características do DER. Outros quatro atributos possuem correlação próxima de zero, são eles: NCOpc (número de colunas opcionais, NCRqr (número de colunas requeridas), NRUna (número de relacionamentos de grau um) e NRNary (número de relacionamentos de grau igual ou superior a 4). Os atributos de opção e de relacionamentos

especiais são poucos explorados no intervalo de dados selecionados. A frequência dos atributos NRUna, NRTer, NUK, NRNAry, e NCMsc é pequena em relação às demais (ver Figura 6-11).

As medidas de correlação e frequência fornecem bons parâmetros para definir o conjunto de atributos que deve fazer parte do modelo. Não basta um atributo ter um elevado número de frequência (ver NCOpc na Figura 6-11), deve ter também alguma importância para produzir um bom modelo (ver NCOpc na Figura 6-10). Os atributos NR (número de relacionamentos) e NFK (número de chaves estrangeiras) são perfeitamente colineares, ou seja, têm uma correlação igual a um. A perfeição colinear ocorre quando duas ou mais variáveis independentes estão fortemente relacionadas linearmente entre si. No caso de modelos preditivos, elas devem ser evitadas (Hirao et al., 2007), por isso, apenas uma das duas será mantida.

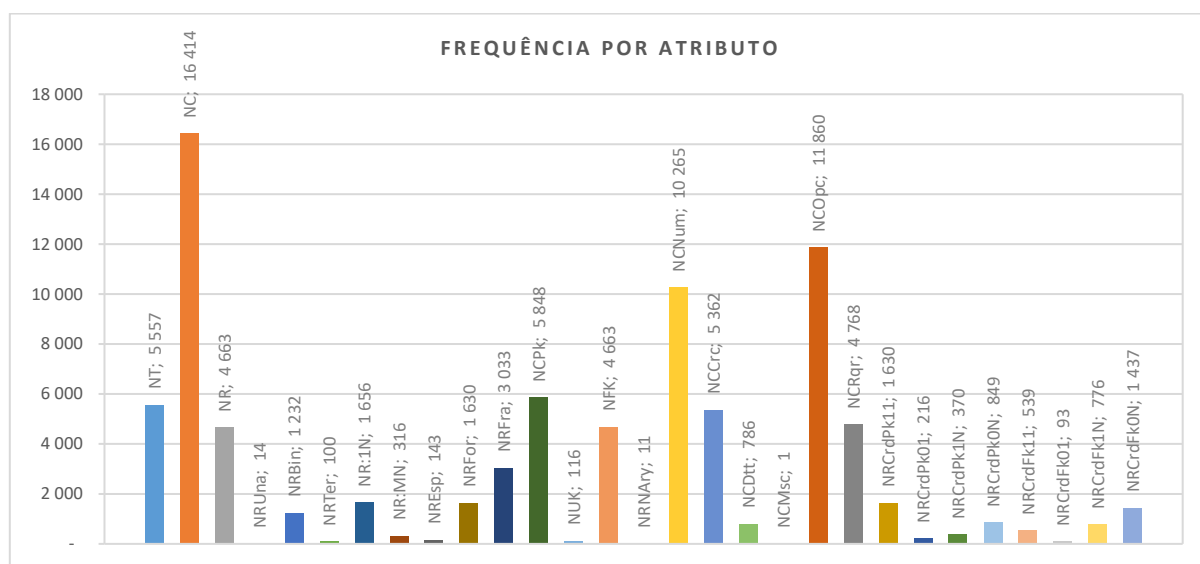


Figura 6-11. Frequência dos atributos no 1º ciclo.

Todos os atributos expõem as características dos DER, e deveriam fazer parte do primeiro ciclo de ensaio. Entretanto, a amostra de dados evidencia que nem se quer foram utilizados todos os atributos, ou que têm pouca ou nenhuma associação com a avaliação do professor, como no caso dos atributos NCOpc e NCMsc. Por conseguinte, alguns dados foram rejeitados. Após os procedimentos de limpeza, transformação e derivação, os seguintes atributos foram selecionados para fazer parte do primeiro ciclo de ensaio: NT, NC, NR, NRBin, NRTer, NR:1N, NR:MN, NREsp, NRFor, NRFra, NCPk, NCNum, NCCrc, NCDtt, NCRdPk11, NCRdPk01, NCRdPk1N, NCRdPk0N, NCRdFk11, NCRdFk01, NCRdFk1N, NCRdFk0N.

Os dados estão desequilibrados, e o total de ocorrências absolutas da classe 1 (falhou) é duas vezes superior às restantes classes de avaliação (ver Figura 6-12). Esse comportamento

nos dados do LabDER ocorre, basicamente, porque o sistema permite ao aluno submeter várias respostas e, conseqüentemente, existe um maior número de submissões de DER do tipo da classe 1, que são as soluções de DER classificadas como falhou. Esse entendimento do funcionamento do ambiente de trabalho explica a nossa abordagem de desequilíbrio dos dados, o que é fundamentado na extração proporcional dos dados por categorias.

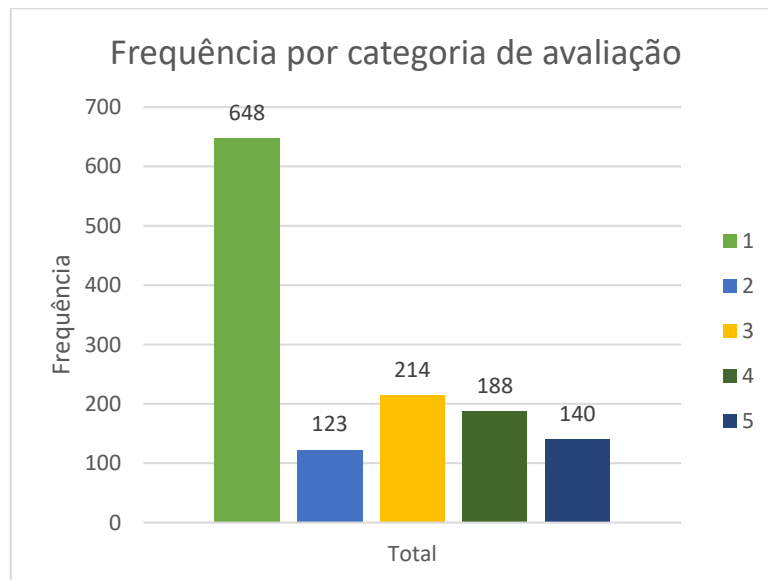


Figura 6-12. Frequência das cinco categorias de avaliação no 1º Ciclo.

A solução para o problema acima mencionado é baseada no algoritmo de estratificação proporcional de dados. Este algoritmo, desenvolvido em Python³, funciona mantendo a mesma proporção de amostras de cada classe no conjunto de dados de treino e teste. Por exemplo, se todo o conjunto de dados tiver dois rótulos (por exemplo, positivo e negativo) e estes tiverem uma proporção de 30/70 e se dividirem em 10 subamostras, cada subamostra estratificada deverá manter a mesma proporção. O uso desta estratégia normalmente torna os modelos mais estáveis para as subamostras (Kohavi, 1995). A fim de provar que o modelo não é enviesado, a medida *Receiver Operating Characteristic* (ROC) é visualizada para todas as classes em cada experiência.

6.3.1.4 Modelação e Avaliação

Ao todo, dez algoritmos de classificação mais populares foram incluídos para construir os modelos preditivos, são eles: *AdaBoostClassifier*⁴ (ABC), *Decision Tree Classifier*⁵ (DTC),

³ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

⁴ <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>

⁵ <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

Gaussian Naive Bayes⁶ (GNB), K-Nearest Neighbors⁷ (K-NN), Linear Discriminant Analysis⁸ (LDA), Logistic Regression⁹ (LR), Multi-layer Perceptron Classifier¹⁰ (MLP), Random Forest Classifier¹¹ (RFC), Stochastic Gradient Descent¹² (SGD), Support Vector Classification¹³ (SVC), que seguiram a configuração descrita no Anexo K: Configurações para a conceção do modelo de aprendizagem. Esta configuração foi obtida por meio de testes empíricos até obter os melhores parâmetros. Um método *grid-search*¹⁴ por otimização de *hyper-parameters* foi aplicado e apresentou resultados semelhantes.

Os algoritmos foram executados usando a biblioteca scikit-learn, um repositório aberto de recursos de aprendizagem computacional, desenvolvido em Python, que disponibiliza vários algoritmos de classificação, regressão e agrupamento (Pedregosa et al., 2011).

O processo de definição do melhor modelo é empírico, por isso foi incluído o maior número de algoritmos possível, no intuito de comparar os resultados de desempenho entre si, e selecionar o melhor. Ainda, através desta experiência, pode confirmar-se, ou não, o que a literatura tem vindo a evidenciar sobre a superioridade dos algoritmos de aprendizagem computacional (por exemplo, redes neurais artificiais, máquinas de vetores de suporte e árvores de decisão) e aos seus equivalentes estatísticos (por exemplo, regressão logística e análise discriminante). Isto porque são menos limitados por suposições e produzem melhores resultados de previsão (Delen, 2010; Delen, Sharda, & Kumar, 2007; Delen, Walker, & Kadam, 2005; Kiang & Y., 2003; Nguyen Thai Nghe, Janecek, & Haddawy, 2007; Sharda & Delen, 2006).

Uma abordagem experimental foi concebida para combinar os dados de treino e teste em nove formatos inversamente proporcionais (ver Figura 6-15), visando descobrir o melhor desempenho dos modelos de classificação. Na primeira fase experimental, os dados de teste possuem 10%, e os dados de treino 90%, na segunda fase 20% de teste e 80% de treino e, assim por diante, até à nona fase de 90% de teste e 10% de treino (ver Figura 6-13).

⁶ https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

⁷ <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

⁸ <https://scikit-learn.org/0.16/modules/generated/sklearn.lda.LDA.html>

⁹ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

¹⁰ https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

¹¹ <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

¹² https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

¹³ <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

¹⁴ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

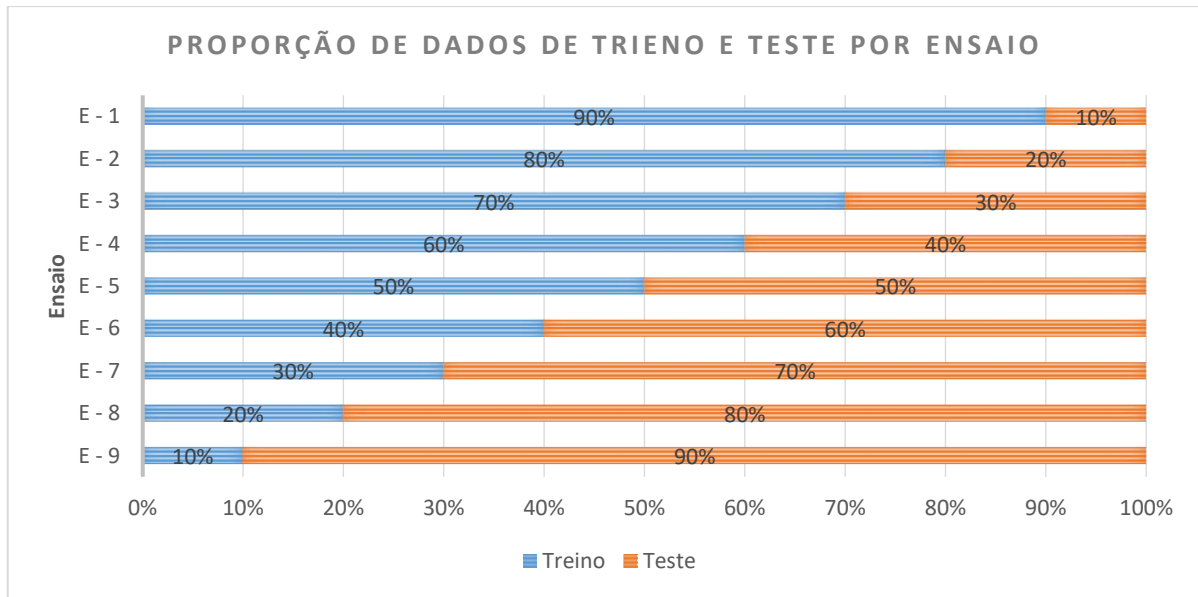


Figura 6-13. Proporção de dados de formação e teste em nove ensaios.

Em cada fase experimental é aplicada uma abordagem de validação cruzada *k-fold* de 10 vezes, que divide os dados em subconjuntos mutuamente exclusivos (ver Figura 6-14).

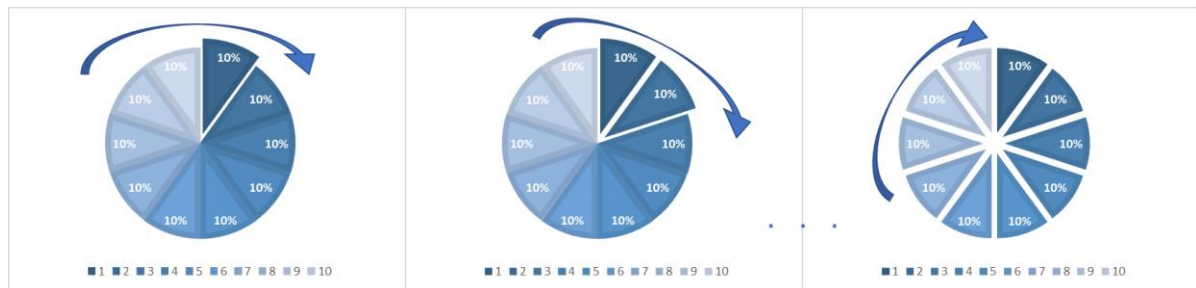


Figura 6-14. Representação gráfica do procedimento de validação cruzada de dez vezes.

A partir de estudos empíricos, tem sido demonstrado que uma divisão de um conjunto de dados em 10 subconjuntos mutuamente exclusivos minimiza o desvio e a variação associada ao processo de validação do modelo (Delen, 2010; Guimarães, Pinto, Nobre, & Zárate, 2018; Kohavi, 1995). Na validação cruzada de 10 vezes, todo o conjunto de dados é dividido em 10 subconjuntos mutuamente exclusivos. Cada conjunto é usado uma vez para testar o desempenho do modelo de classificação que é gerado a partir dos dados combinados dos nove conjuntos restantes, resultando em dez estimativas de desempenho independentes para cada ensaio.

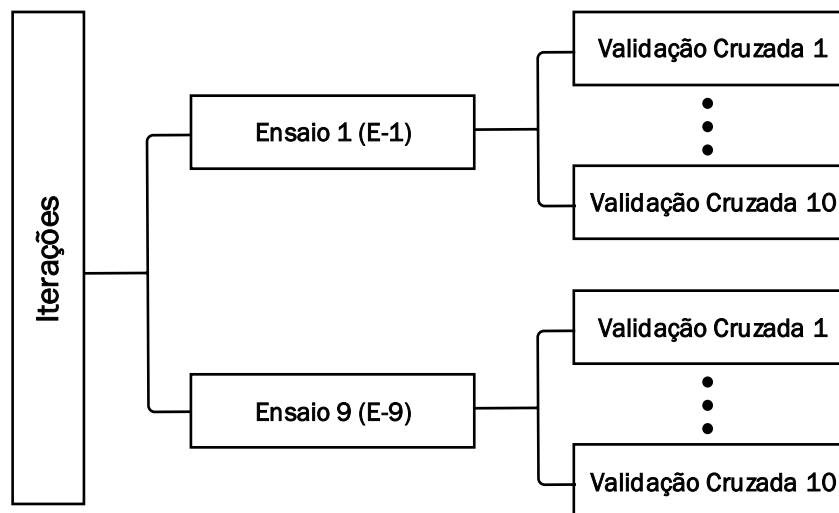


Figura 6-15. Abordagem experimental de nove ensaios de modelos preditivos.

Esta abordagem experimental (ver Figura 6-15) é aplicada em dez algoritmos de classificação (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD, e SVC). Se a validação cruzada for ajustada para 10, então dez modelos diferentes são construídos e testados para cada algoritmo. Ao combinar nove ensaios aplicados ao conjunto de dados originais (isto é, não equilibrados), usando os dez tipos de modelos individuais, um total de 900 modelos são construídos e testados.

Os modelos são avaliados nas medidas de desempenho utilizadas na aprendizagem computacional, como *precision*, *recall*, *F1-Score*, *accuracy*, ROC e AUC, com base na matriz de confusão¹⁵ adaptadas para multiclasse. Para estimar essas medidas, inicialmente os rótulos de teste de cada classe foram codificados usando a técnica *one-hot encoding*. A título de exemplo¹⁶, o processo de conversão de multiclasse em binário para um conjunto de 5 classes (1,2,3,4,5), considera-se uma classe com o valor binário 1 e as restantes com o 0. Este processo repete-se para todas as classes, i.e., considera-se primeiro a classe 1 com o valor 1 e restantes com 0, depois a classe 2 com o valor 1 e restantes com 0, e assim sucessivamente. As probabilidades dos modelos preditivos foram estimadas usando os dados de teste e as medidas de desempenho calculadas por meio das funções da biblioteca do scikit-learn¹⁷¹⁸¹⁹²⁰.

A partir das definições descritas acima, um estudo de caso foi executado com 1.313 dados com dez algoritmos de classificação previamente definidos, usando uma abordagem com nove

¹⁵ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

¹⁶ https://scikit-learn.org/0.15/auto_examples/plot_roc.html

¹⁷ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

¹⁸ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

¹⁹ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html

²⁰ <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html>

ensaios, com uma validação cruzada de 10 vezes. Sendo calculadas, então, as suas medidas de desempenho (ver Figura 6-16).

A Figura 6-16 ilustra um gráfico de desempenho de *precision*, *recall*, *F1-Score*, *accuracy* de 10 algoritmos (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD, SVC). O algoritmo RFC é o mais eficiente, com desempenhos aproximado de 77% para *precision*, *recall*, *F1-Score*, *accuracy*. O segundo algoritmo mais eficiente é o MLP, com cerca de 75% para as quatro métricas de desempenho. Seguido do algoritmo SVC, como terceiro melhor algoritmo para este conjunto de dados, com cerca de 74% de *precision* e 75% para *recall*, *F1-Score* e *accuracy*. Os três algoritmos com o menor desempenho são GNB, ABC e SGD, sendo o GNB com *precision* de 59%, *recall* de 45%, *F1-Score* de 46%, e *accuracy* de 45%, seguido pelo algoritmo ABC com desempenho de 60%, 56%, 57%, e 56% para as quatro métricas, e o algoritmo SGD com desempenho de 61%, 60%, 57%, e 60% para as quatro métricas.

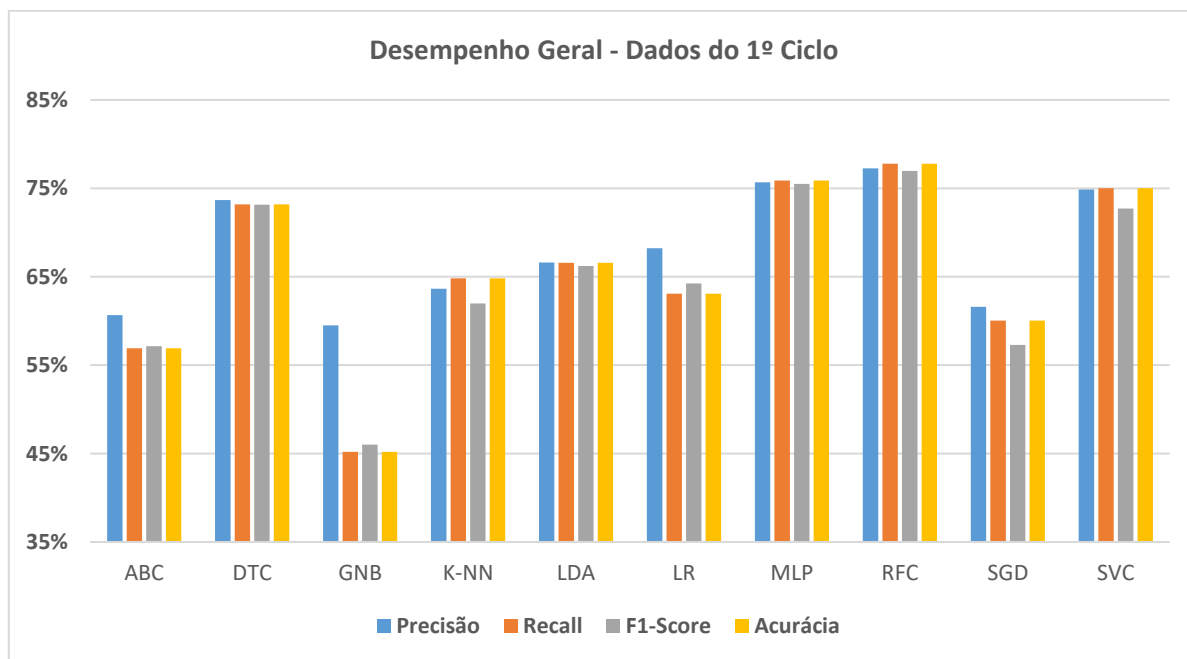


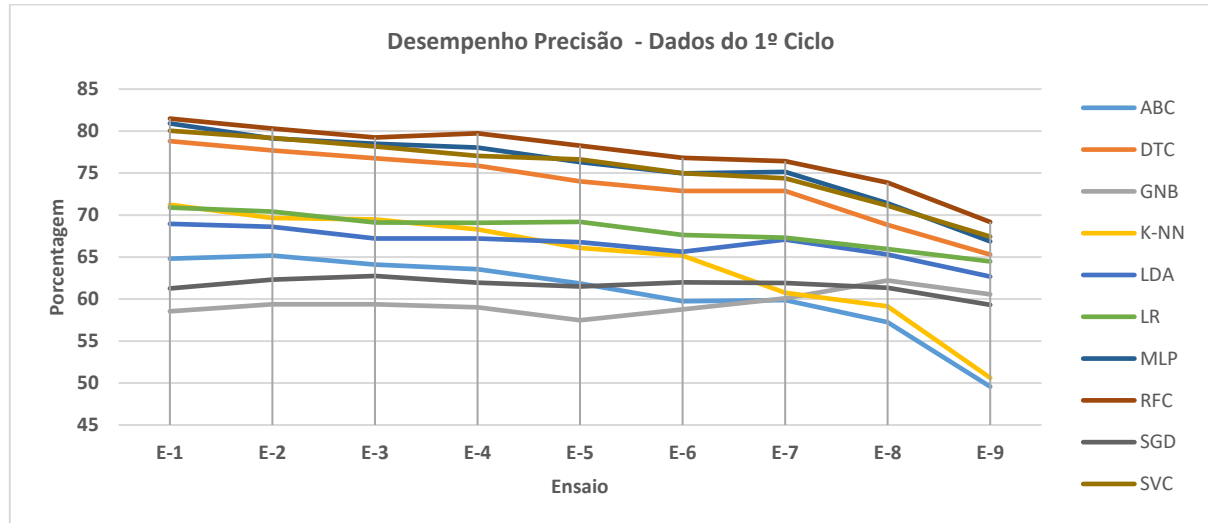
Figura 6-16. Gráfico de comparação de desempenho (*precision*, *recall*, *F1-Score*, e *accuracy*) por algoritmo de classificação (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD e SVC) com os dados do 1º ciclo.

A Tabela 6-6 apresenta o desempenho geral com as taxas de desempenho de *precision*, *recall*, *F1-Score*, e *accuracy* com o desvio padrão para cada um dos dez classificadores anteriormente definidos. Os algoritmos LR, LDA, e SGD possuem desempenhos de *precision*, *recall*, *F1-Score*, e *accuracy* próximo de 15% a menos do que aos desempenhos do algoritmo RFC. Porém, os desvios-padrão são bem inferiores em relação aos demais algoritmos de classificação. Um dos critérios de sucesso estabelecidos é de que o algoritmo deve possuir uma variação de no máximo 5% de desvio padrão.

Tabela 6-6. Comparação do desempenho dos algoritmos com os dados do 1º ciclo.

Algoritmo	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
ABC	60,66 ±4,94	56,90 ±5,75	57,15 ±5,94	56,90 ±5,75
DTC	73,67 ±4,36	73,18 ±4,80	73,16 ±4,62	73,18 ±4,80
GNB	59,49 ±1,35	45,20 ±0,92	46,00 ±1,49	45,20 ±0,92
K-NN	63,65 ±4,36	64,82 ±3,58	61,97 ±4,00	64,82 ±3,58
LDA	66,60 ±2,09	66,59 ±2,34	66,22 ±2,14	66,59 ±2,34
LR	68,23 ±2,09	63,08 ±1,45	64,24 ±1,43	63,08 ±1,45
MLP	75,69 ±4,33	75,88 ±4,21	75,50 ±4,28	75,88 ±4,21
RFC	77,25 ±3,81	77,79 ±3,34	76,96 ±3,65	77,79 ±3,34
SGD	61,59 ±0,98	60,05 ±1,23	57,29 ±1,11	60,05 ±1,23
SVC	74,87 ±3,91	75,02 ±4,30	72,71 ±5,46	75,02 ±4,30

A Figura 6-17 ilustra o gráfico comparativo do desempenho *precision* dos classificadores para cada um dos nove ensaios. Aqui, os dados de teste e treino variam entre o E-1 com 10% de teste e 90% de treino até o E-9 com 90% de teste e 10% de treino. Mesmo com um conjunto de dados variável, alguns algoritmos mantêm-se com o mesmo desempenho ao longo dos ensaios. Isso significa que os atributos fornecem um conjunto de características muito importantes para os algoritmos aprenderem, de forma equivalente, com uma variação grande entre os dados de treino e de teste.

Figura 6-17. Gráfico de *precision* para os dados do primeiro ciclo.

A Figura 6-18 ilustra o gráfico de comparação do desempenho *recall* para os nove ensaios. O *recall* faz parte do critério de sucesso e, a partir dessa medida, será definido o algoritmo que oferece classificadores com a menor taxa de erro no contexto da aplicação LabDER. Do algoritmo RFC resultam os melhores classificadores, seguido pelos algoritmos SVC, MLP e DTC. Por outro lado, os algoritmos GNB, ABC, e SGD fornecem os modelos com menor desempenho. Novamente, pode ser observado o comportamento padrão dos

algoritmos LR e LDA na construção dos classificadores para os diferentes tipos de dados, ao longo do ensaio.

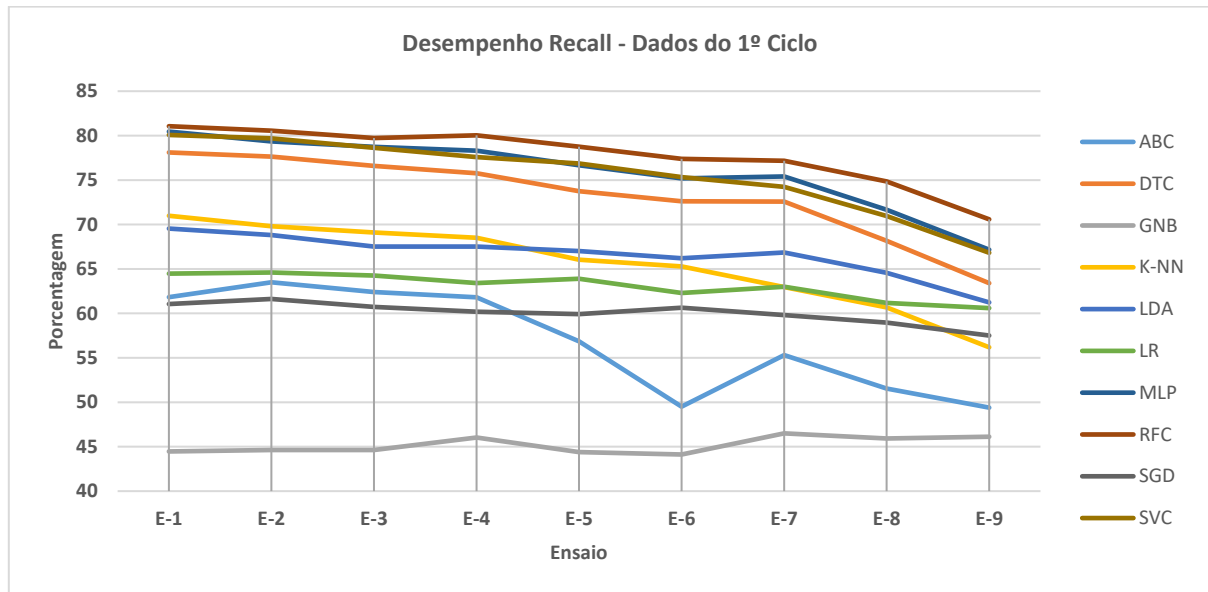


Figura 6-18. Desempenho do *recall* para os dados do primeiro ciclo.

A Figura 6-19 ilustra o gráfico de comparação do desempenho F1-Score para os nove ensaios. Essa medida deve ser interpretada como a medida global do classificador, uma vez que combina os desempenhos de *precision* e *recall* de modo a trazer um número único que explique a qualidade geral do modelo. Dois algoritmos RFC e DTC resultam classificadores com *recall* equivalentes no E-1, mas esse comportamento não ocorre no E-9. O algoritmo GNB é totalmente inapropriado a este contexto de projeto, provocado por uma taxa elevada de *precision* e uma baixa taxa de *recall* e, conseqüentemente, menor taxa de *F1-score*.

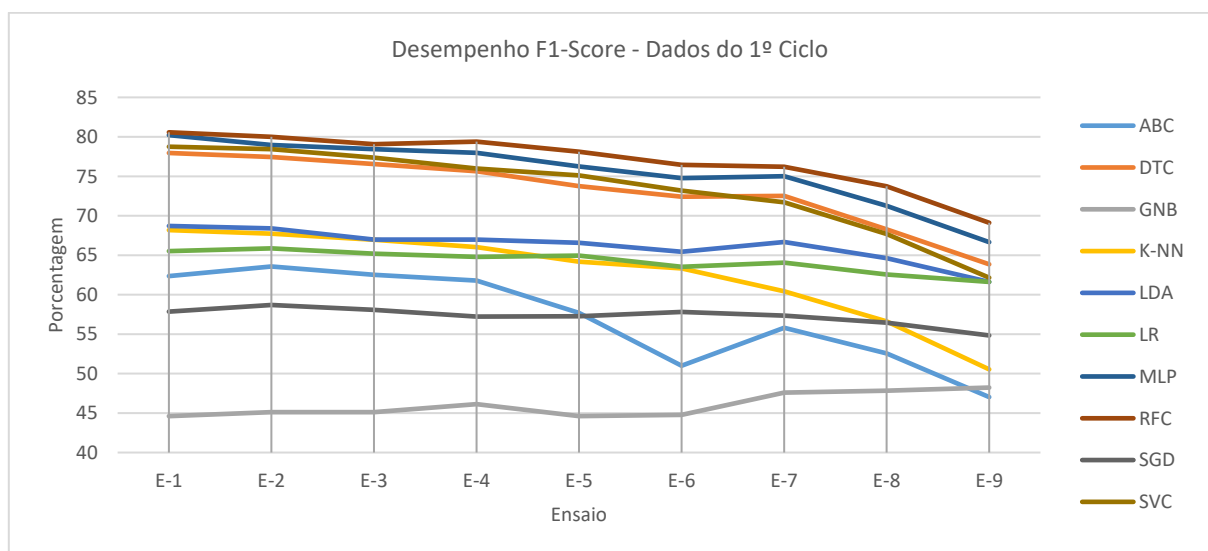


Figura 6-19. Desempenho do *F1-Score* para os dados do primeiro ciclo.

Por fim, e não menos importante, o desempenho de *accuracy* é ilustrado na Figura 6-20 para os nove ensaios. Essa medida descreve, no geral, com que frequência o classificador está correto, uma vez que tanto os verdadeiros positivos como os verdadeiros negativos são processados para gerar o índice geral do modelo.

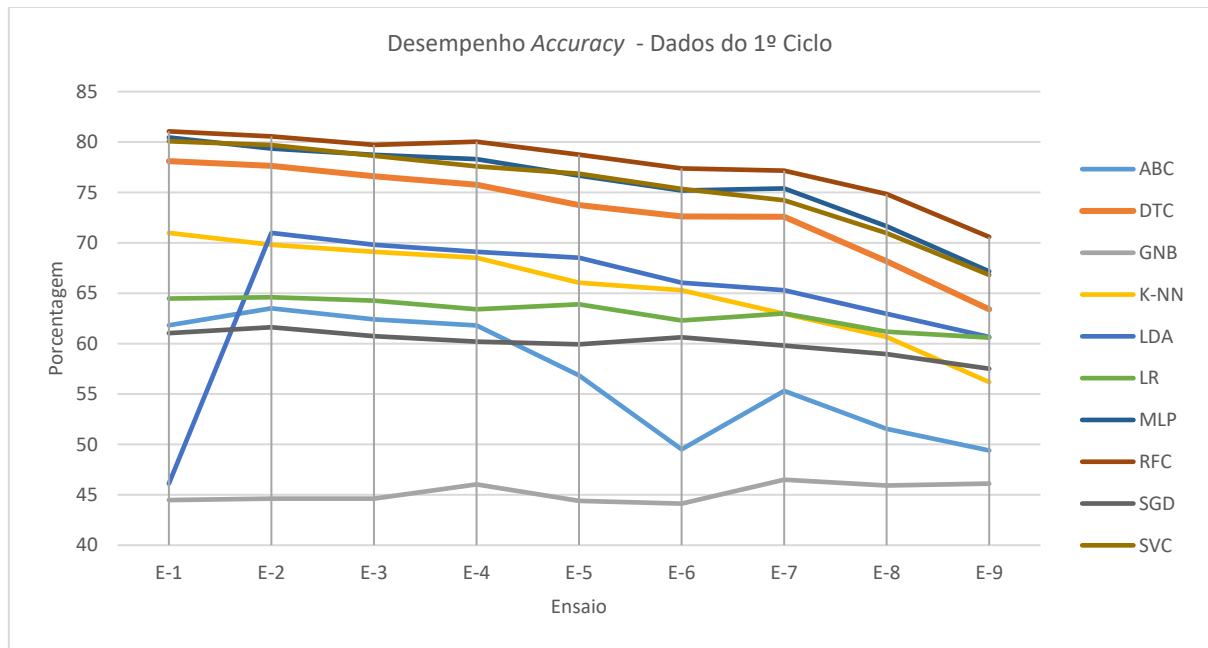


Figura 6-20. Desempenho de *accuracy* para os dados do primeiro ciclo.

Por isso, essa medida está incluída no objetivo deste projeto, com uma taxa mínima de 90%. Três algoritmos RFC, MLP, e SVC resultam classificadores com maior *accuracy*, entre 75% a 83%. Por outro lado, os três algoritmos dos quais resultam modelos com *accuracy* entre 40% e 60% são SGD, ABC e GNB. Entretanto, o algoritmo LDA teve a *accuracy* menor no E-1 e mostrou-se pouco estável, mesmo num conjunto de dados de treino tão elevados.

Os resultados obtidos indicam que os dez algoritmos utilizados não conseguem gerar modelos com uma *precision* acima de 90% com o conjunto de 1.313 dados selecionados para este ciclo. Mais, os resultados indicam que os algoritmos também não conseguem atender à meta de 10% com os dados utilizados. Assim, um novo ciclo do CRISP-DM deve ser executado, procurando, inicialmente, descobrir os problemas nos dados e possíveis soluções, passíveis de serem implementadas, com o intuito de encontrar um modelo que satisfaça os critérios de sucesso definidos.

6.3.2 Segundo ciclo

Os resultados de *accuracy* obtidos no primeiro ciclo são inferiores aos exigidos. Este ciclo visa identificar os problemas que levaram ao desempenho obtido e a serem aplicadas alternativas, na tentativa de obter um modelo com no mínimo 90% de *accuracy*.

6.3.2.1 Entendimento do Negócio

Neste ciclo, pretende-se descobrir as razões pelas quais os modelos preditivos não conseguiram dar uma *accuracy* satisfatória. Os algoritmos de aprendizagem computacional supervisionada são métodos eficientes para construir modelos preditivos a partir de padrões detetados nos dados de entrada e saída. Assim, é necessário responder às seguintes questões:

- 1) Quais as classes que são mais classificadas de forma incorreta?
- 2) Existem avaliações ambíguas? Ou seja, os professores atribuem diferentes classificações a um mesmo tipo de resposta DER?
- 3) Se sim na questão 2, qual deveria ser a classificação correta?

Para responder à primeira questão, o primeiro passo é analisar o AUC e a matriz de confusão de um dos resultados obtidos do primeiro ciclo (ver Figura 6-21). Nota-se no gráfico AUC, pela matriz de confusão, que as classes 2, 3 e 4 são as que têm maior erro de classificação.

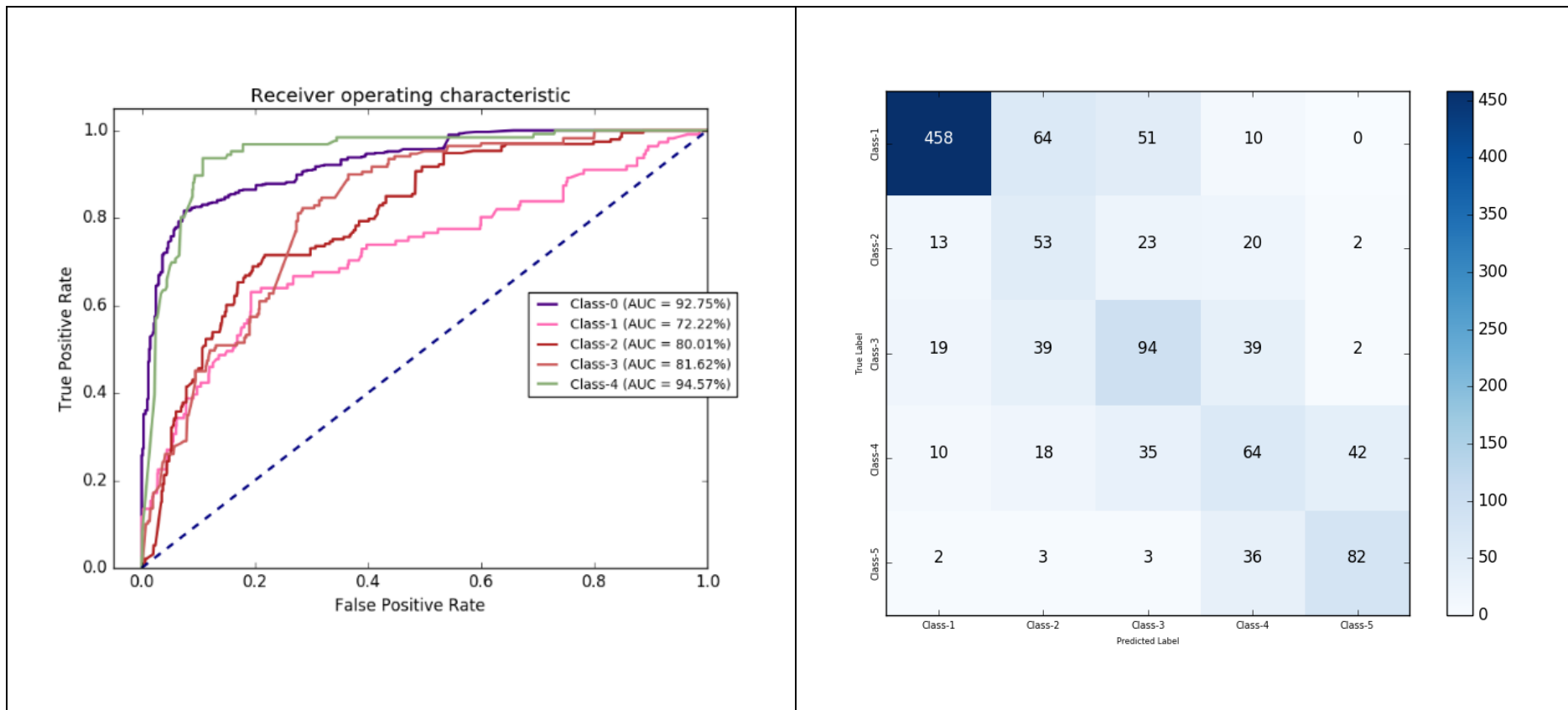


Figura 6-21. Captura das medidas de desempenho AUC e matriz de confusão de forma aleatória no 1ª ciclo.

Para responder à segunda questão, foram identificadas as questões de bases de dados do LabDER classificadas nas classes 2, 3 e 4 com os mesmos conjuntos de métricas, ou seja, questões com diferentes critérios de avaliação e com o mesmo número de componentes de DER na resposta do aluno. Ilustramos na Figura 6-22 duas respostas de DER com o mesmo número de componentes de DER, às quais foram atribuídas classificações diferentes, pelo mesmo professor.

Para responder à terceira questão, foram apresentados aos professores os casos em que existia uma maior ocorrência deste tipo de avaliação. Após algumas reuniões com os professores, foi identificada a dificuldade destes em atribuir a classificação, principalmente entre as categorias 2 e 3. Foi recomendado aos professores que definissem um conjunto de critérios objetivos, baseados no *feedback* da ferramenta, onde identificassem a categoria de avaliação a partir da quantidade de erros cometidos por conceitos de DER. Após algumas reuniões, os professores decidiram adotar um critério de avaliação (ver Tabela 6-7) que penalizasse cada conceito de DER errado.

Tabela 6-7. Critério de avaliação do DER para novas avaliações de DER.

Itens de Avaliação	Valor a descontar
1) Domínio do atributo errado	0,25
2) Colunas em excesso ou em falta	1
3) Tabelas em excesso ou em falta	2
4) Chaves primárias em excesso ou em falta	1
5) Relações em excesso ou em falta	1
6) Chaves estrangeiras ligadas a campos incorretos	1
7) Chaves estrangeiras sem cardinalidade	2
8) Chaves estrangeiras com cardinalidades incorretas	0,25
9) Chaves estrangeiras inversamente conectadas	0,25
10) Conceitos de entidade forte e fracas erradas	1
Nota máxima "5" apenas quando tudo estiver correto O peso está relacionado com a quantidade de itens a serem avaliados. O peso do erro é maior quanto menos objetos de DER a avaliar. Toda a resposta começa com nota máxima "5", subtrai-se então cada um dos erros.	

269-Suponha que você tenha o ERM mostrado na Figura Q3.14. Como você converteria este modelo em um ERD que exibe apenas relacionamentos 1: M? (Certifique-se de criar o ERD.)

269-Suponha que você tenha o ERM mostrado na Figura Q3.14. Como você converteria este modelo em um ERD que exibe apenas relacionamentos 1: M? (Certifique-se de criar o ERD.)

Alterar Nota Nota=3 (50% - 69%) - Suficiente Máx.Pontos:5.00

Alterar Nota Nota=4 (70% - 89%) - Bom Máx.Pontos:5.00

Q.268	Q.269	Q.270	Q.271	Q.272	Q.273	Q.274	Q.27:
5.00	1.00	3.00	2.00	2.00	5.00	2.00	1.00
5.00	1.00	3.00	2.00	3.00		5.00	1.00
2.00	3.00	3.00	3.00	3.00		5.00	1.00
4.00	4.00	3.00	5.00	3.00			3.00
3.00	4.00	4.00		3.00			3.00
1.00	4.00	4.00		3.00			3.00
...

Figura 6-22. Exemplo de classificações diferentes para respostas de DER com mesmo número de componentes de DER, lado esquerdo 3 e lado direito 4.

Assim, a resposta da terceira questão, “qual deve ser a resposta correta?”, foi decidida pelos professores a partir das regras que eles definiram. Como exemplo, ao aplicar os critérios de avaliação de estabelecidos na Tabela 6-7, usando a resposta do professor (ver Figura 6-23) no problema anterior (ver Figura 6-22), o resultado seria 3.

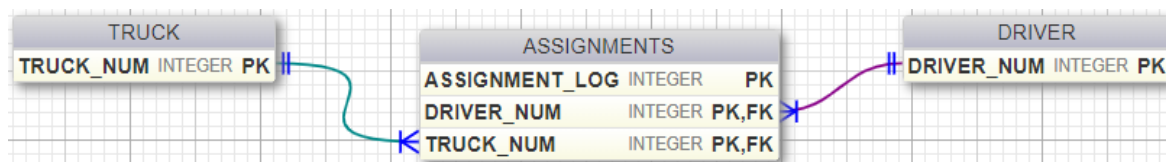


Figura 6-23. Resposta do professor.

Desta forma, algumas respostas dos alunos foram reavaliadas pelos professores, o que implicou a eliminação de um conjunto de dados não reavaliados e a recolha de mais dados, de forma a obter uma quantidade suficiente para executar um novo ensaio. Nesse sentido, o LabDER recebeu atualizações, e foi incluído o *feedback* semântico sem nota, para incentivar os alunos a submeterem mais respostas e a ser obtida a variabilidade de dados necessária. Além disso, os professores receberam instruções da nova avaliação e, à partida, a sua utilização evitaria que voltasse a ocorrer o problema detetado nos dados.

6.3.2.2 Compreensão dos Dados

O total de dados selecionados para este segundo ciclo são de 4.712 registos, que foram recolhidos nas instituições do ISCAC e IPCA após os dados serem atualizados pelos professores conforme descrito na secção anterior. Os atributos a serem utilizados no modelo são os mesmos do primeiro ciclo, ou seja, como entrada tem-se a distância normalizada entre os atributos do DER do aluno e da resposta do professor na escala de 0 e 1 e também as avaliações na escala de 1 a 5 submetidas pelos professores. Todos os dados ficam armazenados na BD do sistema e são processados da mesma forma, sem nenhuma mudança de rotina de sistema.

6.3.2.3 Preparação dos Dados

A preparação dos dados seguiu o mesmo fluxo de desenvolvimento do 1º ciclo, com a seleção dos mesmos atributos com a distância normalizada na escala de 0 e 1, entre a resposta do aluno e a resposta do professor, onde 0 significa que não há diferença entre os DER, e 1 significa que os DER são totalmente diferentes.

A seleção dos atributos também é fundamentada pela análise de correlações e na frequência dos atributos. As correlações das variáveis são descritas no Anexo G: Métricas - Análises de Correlação - 2º Ciclo. A importância dos atributos de entrada é ilustrada na Figura

6-24, isto é, a correlação entre os atributos de entrada (distância dos atributos do DER do aluno e da resposta do professor) e de saída (o atributo de avaliação do professor).

Os atributos têm uma correlação maior quando comparadas com as do primeiro ciclo, entretanto as métricas de NUK (número de colunas únicas), NCMsc (número de colunas variadas), NCOpc (número de colunas opcionais) continuam a ter uma baixa correlação com a variável de saída (ver Figura 6-24).

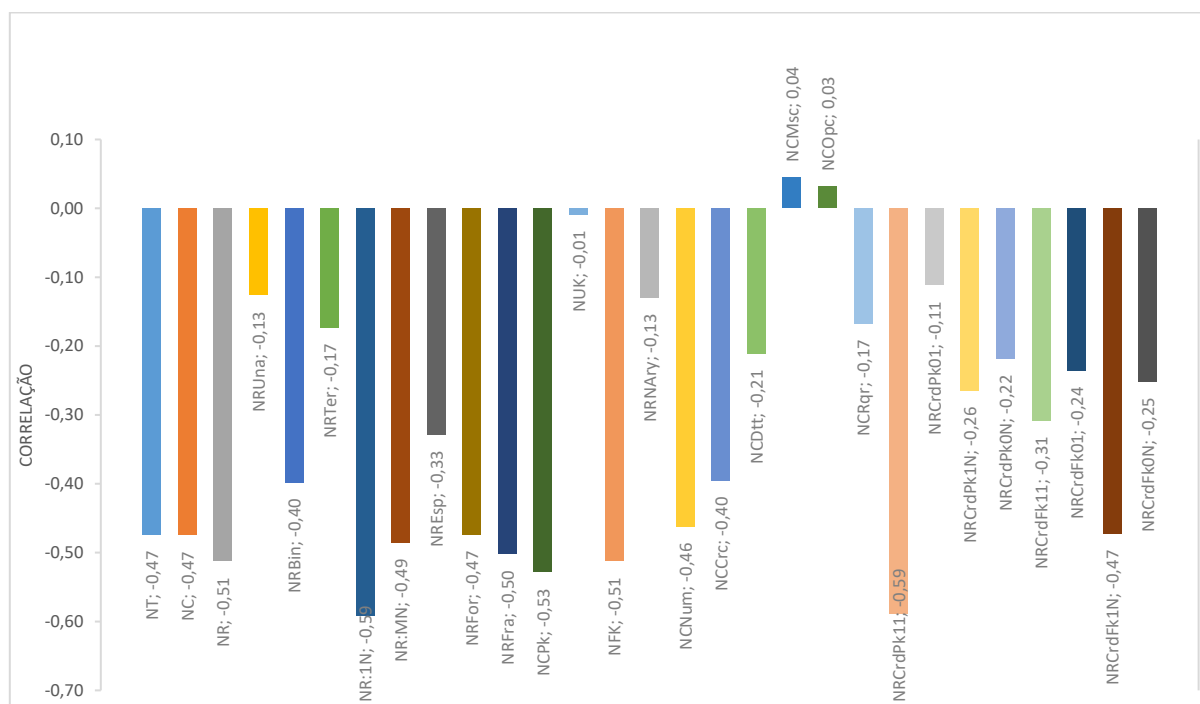


Figura 6-24. Correlação dos atributos de entrada no 2º ciclo.

A Figura 6-25 ilustra uma comparação entre as correlações de cada métrica. As métricas no 2º ciclo têm uma correlação de cerca de 10% superior ao do 1º ciclo. Ao todo 17 atributos aumentaram a correlação (NT, NC, NR, NRUna, NRFor, NRFra, NCPk, NFK, NFK, NRNary, NCNum, NCRqqr, NCRdPk11, NCRdPk01, NCRdPk1N, NCRdFk11, NCRdFk1N, NCRdFk0N). Por outro lado, 12 atributos diminuíram a correlação (NRBin, NRTer, NR:1N, NR:MN, NREsp, NUK, NCCrc, NCDtt, NCMsc, NCOpc, NCRdPk0N, NCRdFk01).

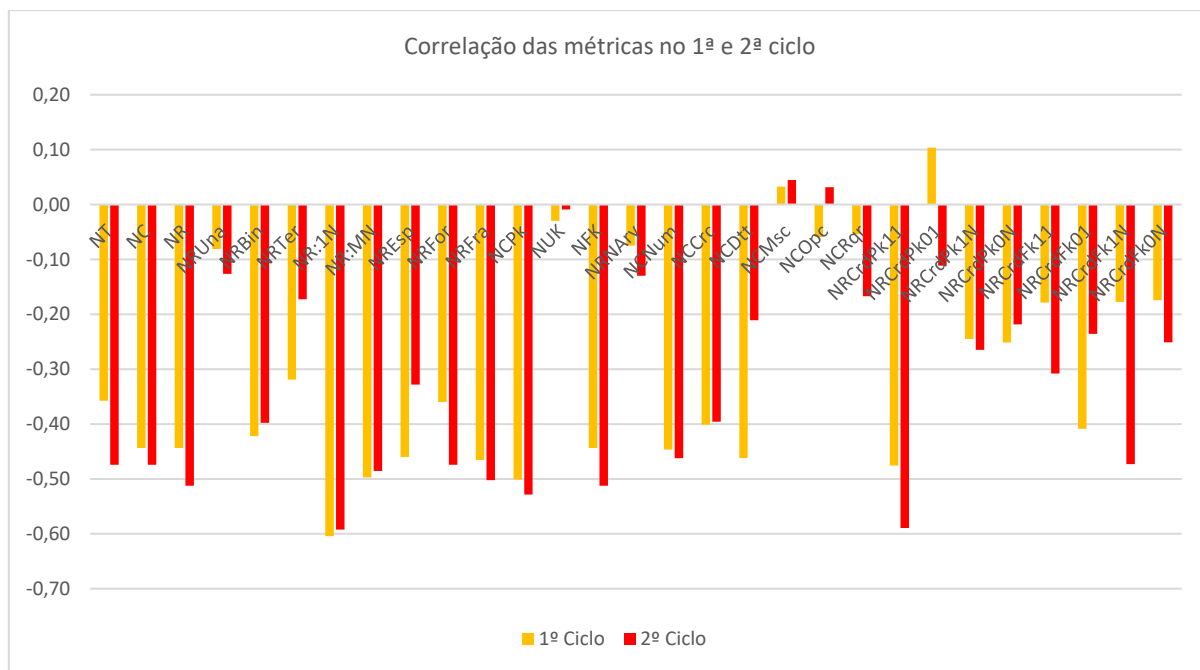


Figura 6-25. Comparação entre a correlação dos atributos no 1º e 2º ciclo.

A frequência dos atributos NRUna, NRTer, e NRNary é muito superior à da frequência do 1º ciclo. Por outro lado, a de NUK e NCMsc são extremamente pequenas em relação às demais (ver Figura 6-26). Repetidamente, a frequência dos atributos NCRqr e NCOpC é muito elevada e tem pouca correlação. Também não foi utilizada nesse 2º ciclo.

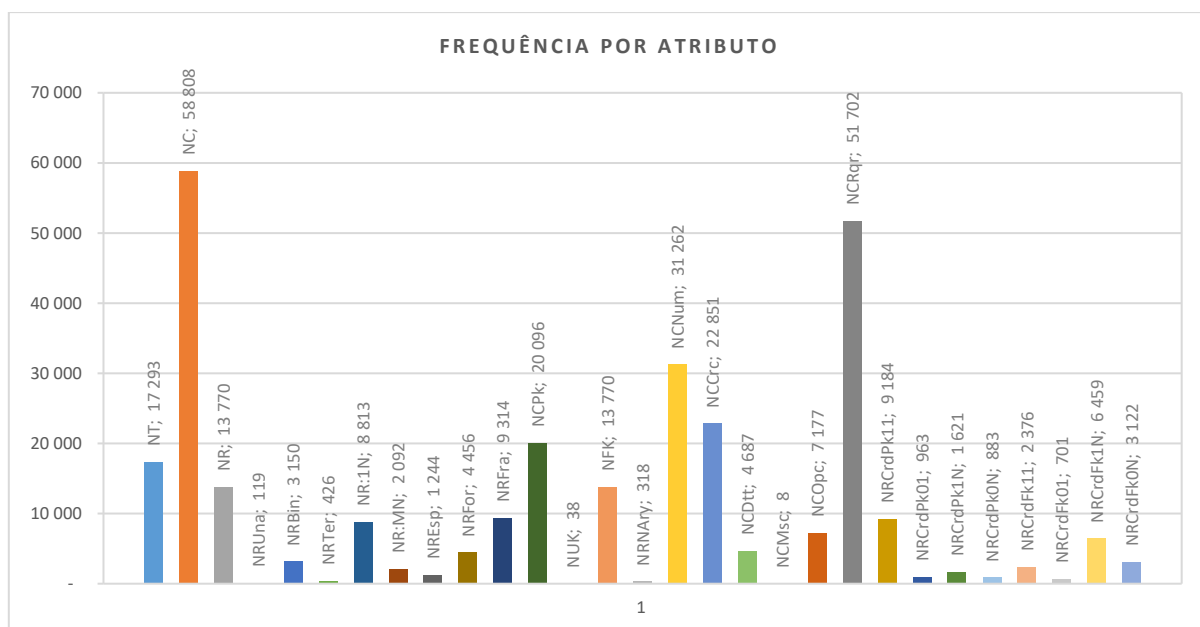


Figura 6-26. Frequência dos atributos no 2º ciclo.

Os atributos que não foram utilizados foram excluídos em conjunto com os atributos com baixa correlação. No final deste processo, alguns dados foram rejeitados e, após os procedimentos de limpeza, transformação e derivação, os seguintes atributos foram

selecionados para fazer parte do segundo ciclo do estudo: NTbl, NCIn, NR, NRUna, NRBin, NRTrn, NRNAry, NR:1N, NR:MN, NREsp, NRFor, NRFra, NCPK, NCNum, NCCrc, NCDtt, NRCrdPk11, NRCrdPk01, NRCrdPk1N, NRCrdPk0N, NRCrdFk11, NRCrdFk01, NRCrdFk1N, NRCrdFk0N.

Os dados estão desequilibrados de forma análoga no 1º ciclo, pelas razões já conhecidas, onde basicamente o sistema ignora os erros e considera sempre a melhor resposta do aluno, e, por esta razão, existem mais dados da primeira categoria. Este segundo ciclo é caracterizado por um volume até quatro vezes maior que o primeiro ciclo (ver Figura 6-27).

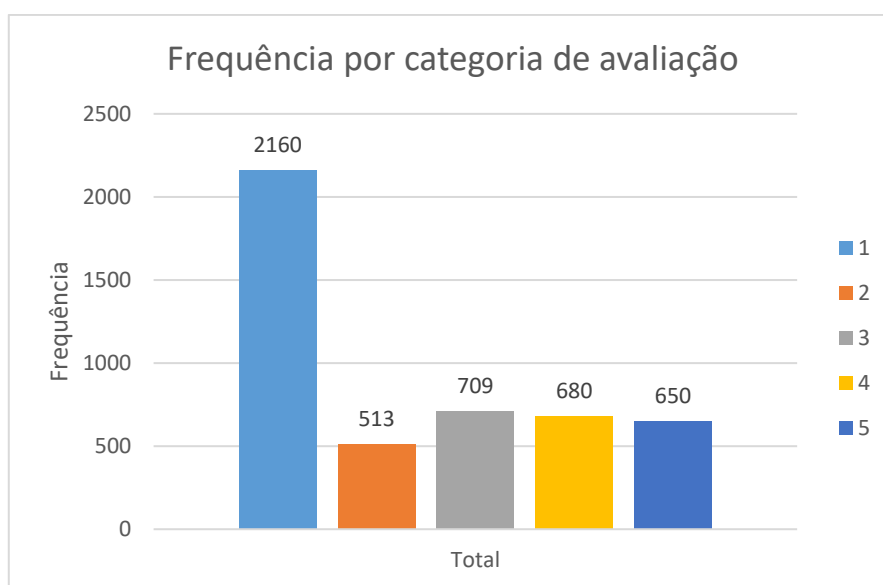


Figura 6-27. Frequência das cinco categorias de avaliação no 2º Ciclo

A solução para o problema dos dados não equilibrados também já foi discutida no primeiro ciclo, tendo sido adotada a solução de estratificação proporcional de dados.

6.3.2.4 Modelação e Avaliação

A modelação seguiu o mesmo processo do primeiro ciclo, e foram usados os mesmos critérios de avaliação de desempenho. Assim, um novo caso de estudo foi executado com 4.712 registos e utilizados em dez classificadores (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD, SVC), neste segundo ciclo de ensaio. Seguindo a abordagem de nove ensaios com validação cruzada de 10 pontos, foram obtidos mais 900 modelos. As informações medidas de desempenho dos modelos foram processadas (ver Anexo I: Resultado dos ensaios - 2º Ciclo), e então comparadas.

O resultado do desempenho do 2º ciclo pode ser visto na Figura 6-28, que ilustra uma comparação de desempenho (*precision*, *recall*, *F1-Score*, e *accuracy*) aplicada aos dez algoritmos anteriormente definidos (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD, SVC). O algoritmo RFC é o mais eficiente e atende a todos os critérios de sucesso, com desempenhos de 90.07% de *precision*, 90.20% de *recall*, 90.06% de *F1-Score* e, 90.20% *accuracy*. O segundo algoritmo mais eficiente é o MLP, com cerca de 89% para as quatro métricas de desempenho. Seguido do algoritmo DTC como o terceiro melhor algoritmo para este conjunto de dados, com cerca de 88% de desempenho para as quatro métricas. Os três algoritmos com o menor desempenho são ABC, GNB e SGD, sendo o ABC com uma *precision* de 50%, *recall* de 44%, *F1-Score* de 40%, e *accuracy* de 44%, seguido pelo algoritmo GNB com desempenho de 59%, 45%, 46%, e 45% para as quatro métricas, e o algoritmo SGD com desempenho de 67%, 67%, 54%, e 67% para as quatro métricas.

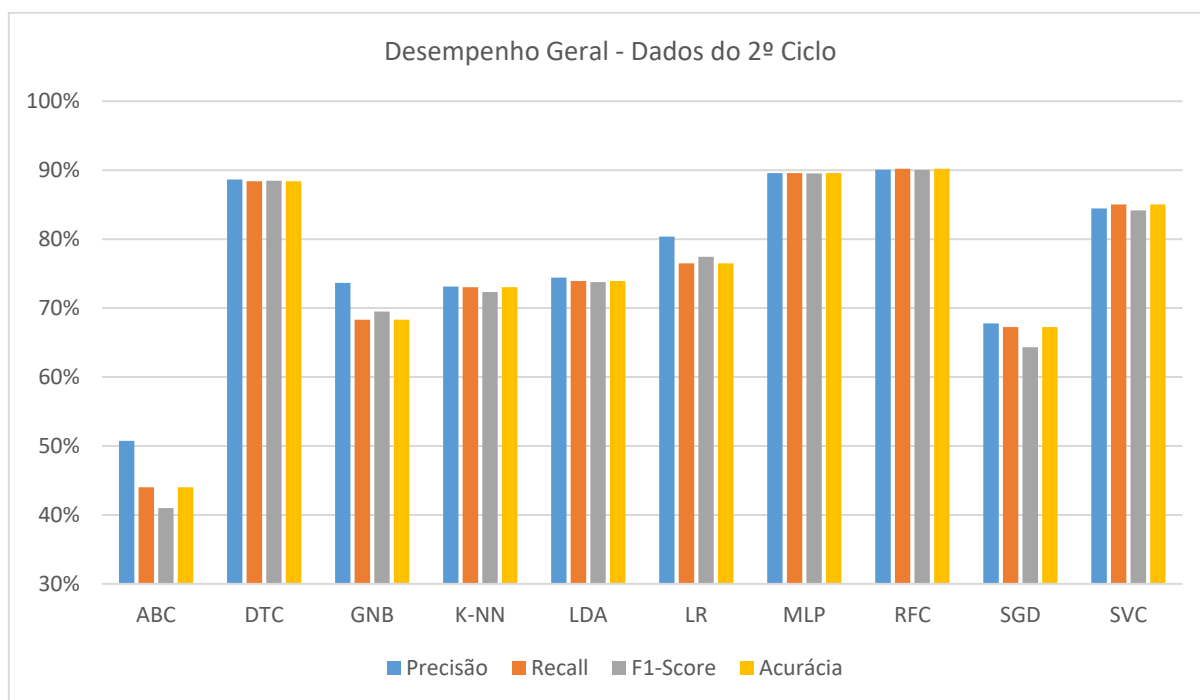


Figura 6-28. Gráfico de comparação de desempenho (*precision*, *recall*, *F1-Score*, e *accuracy*) por algoritmo de classificação (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD e SVC) com os dados do 2º ciclo.

A Tabela 6-8 apresenta uma comparação entre o desempenho geral dos 1º e 2º ciclos com as taxas de desempenho de *precision*, *recall*, *F1-Score* e *accuracy* com o desvio padrão para cada um dos dez classificadores anteriormente definidos. Os modelos do 2º ciclo têm um desempenho de cerca de 25% superior aos dos modelos do primeiro ciclo. Por muito que os modelos GNB não sejam utilizados, destacamos que o algoritmo teve um aumento de quase 40% a mais, neste novo ciclo de ensaios. No geral, o desvio padrão foi reduzido para metade. O algoritmo RFC tem o maior desempenho e atende ao critério mínimo de 90% de *accuracy* e

recall. Os algoritmos LR e SGD têm desempenhos de *precision*, *recall*, *F1-Score* e *accuracy* cerca de 15% inferiores aos desempenhos do algoritmo RFC. Porém, o desvio padrão é o menor entre todos os algoritmos classificadores. Neste ciclo, apenas o algoritmo RFC atende todos os critérios de sucesso estabelecidos.

Tabela 6-8. Comparação do desempenho dos algoritmos com os dados do 1º e 2º ciclo.

<i>Algoritmo</i>	<i>Ciclo</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F1-Score (%)</i>	<i>Accuracy (%)</i>
ABC	1 ^a	60,66 ±4,94	56,90 ±5,75	57,15 ±5,94	56,90 ±5,75
	2 ^a	50,75 ±5,72	44,02 ±5,48	40,98 ±6,22	44,02 ±5,48
DTC	1 ^a	73,67 ±4,36	73,18 ±4,80	73,16 ±4,62	73,18 ±4,80
	2 ^a	88,66 ±3,71	88,39 ±3,91	88,47 ±3,84	88,39 ±3,91
GNB	1 ^a	59,49 ±1,35	45,20 ±0,92	46,00 ±1,49	45,20 ±0,92
	2 ^a	73,65 ±0,81	68,31 ±1,29	69,49 ±1,43	68,31 ±1,29
K-NN	1 ^a	63,65 ±4,36	64,82 ±3,58	61,97 ±4,00	64,82 ±3,58
	2 ^a	73,11 ±3,86	73,02 ±2,96	72,33 ±3,54	73,02 ±2,96
LDA	1 ^a	66,60 ±2,09	66,59 ±2,34	66,22 ±2,14	66,59 ±2,34
	2 ^a	74,43 ±1,38	73,93 ±1,26	73,77 ±1,34	73,93 ±1,26
LR	1 ^a	68,23 ±2,09	63,08 ±1,45	64,24 ±1,43	63,08 ±1,45
	2 ^a	80,36 ±1,82	76,50 ±1,65	77,43 ±1,72	76,50 ±1,65
MLP	1 ^a	75,69 ±4,33	75,88 ±4,21	75,50 ±4,28	75,88 ±4,21
	2 ^a	89,59 ±2,69	89,57 ±2,63	89,55 ±2,66	89,57 ±2,63
RFC	1 ^a	77,25 ±3,81	77,79 ±3,34	76,96 ±3,65	77,79 ±3,34
	2 ^a	90,07 ±2,49	90,20 ±2,37	90,06 ±2,48	90,20 ±2,37
SGD	1 ^a	61,59 ±0,98	60,05 ±1,23	57,29 ±1,11	60,05 ±1,23
	2 ^a	67,77 ±1,48	67,24 ±1,14	64,33 ±1,47	67,24 ±1,14
SVC	1 ^a	74,87 ±3,91	75,02 ±4,30	72,71 ±5,46	75,02 ±4,30
	2 ^a	84,45 ±3,89	85,04 ±3,58	84,17 ±4,24	85,04 ±3,58

A comparação dos desempenhos com os dados do 1º e 2º ciclos é ilustrado no gráfico da Figura 6-29. Todos os algoritmos tiveram um aumento significativo de desempenho de *precision*, *recall*, *F1-Score*. Isso demonstra que os dados do 2º ciclo geram modelos classificadores melhores. Sendo os algoritmos RFC, MLP, e DTC visualmente (ver Figura 6-29) mais eficientes e os ABC, GNB, e SGD, os menos eficientes, estando muito abaixo da meta estabelecida.

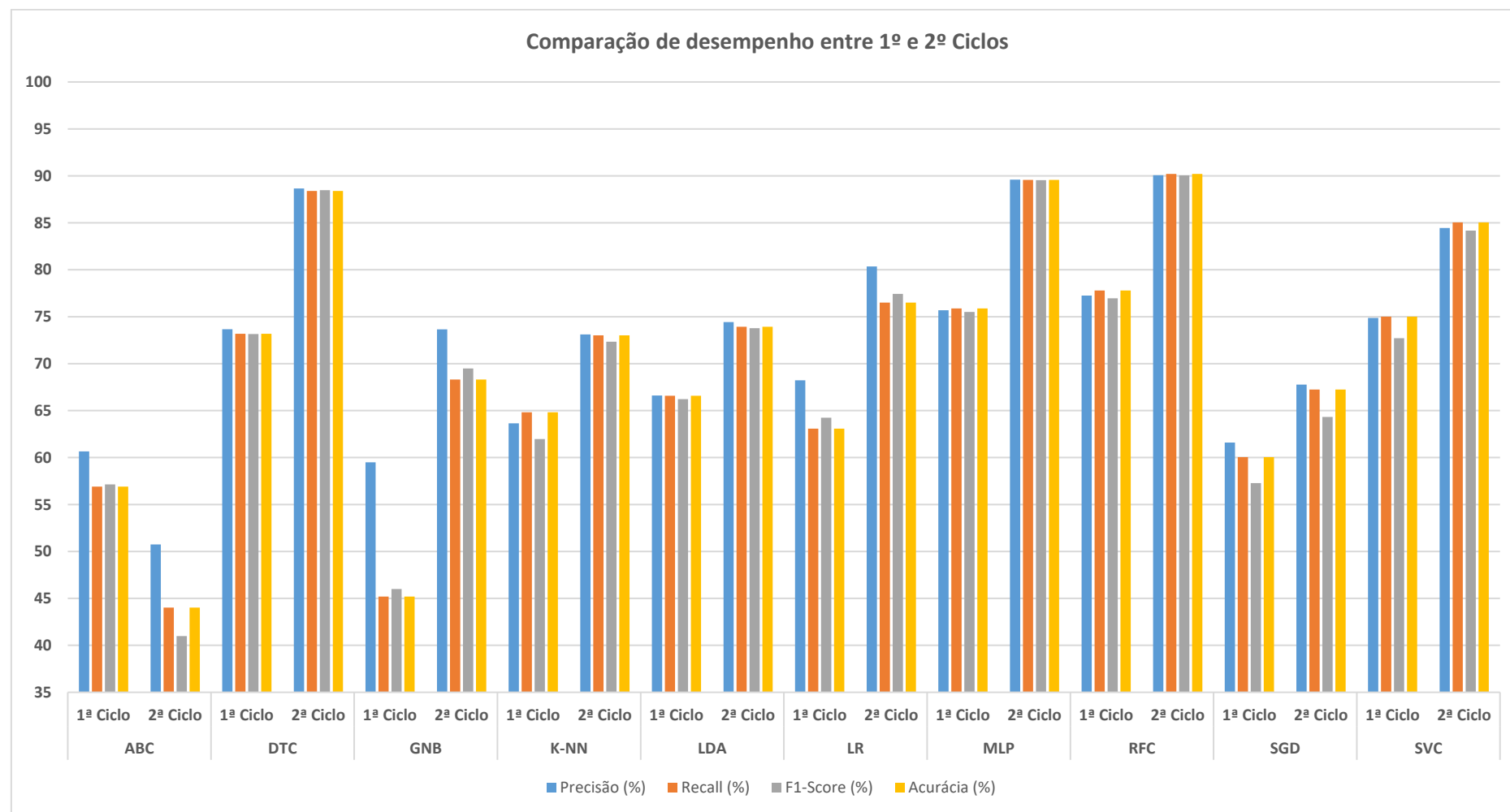


Figura 6-29. Gráfico comparativo de desempenho entre os 1º e 2º ciclos.

A Figura 6-30 ilustra o gráfico comparativo do desempenho de *precision* dos classificadores para cada um dos nove ensaios no 2º ciclo. Os dados deste ciclo fornecem um conjunto de características muito mais representativas para os algoritmos de aprendizagem computacional. À exceção do algoritmo ABC, todos conseguem aprender de forma eficiente entre os nove ensaios.

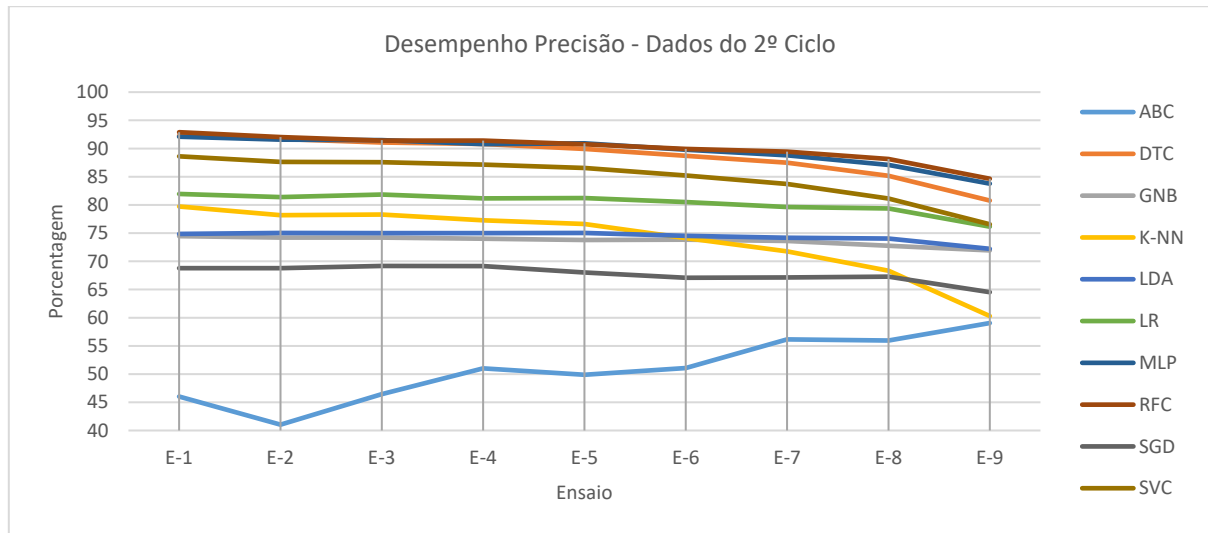


Figura 6-30. Desempenho de *precision* para os dados do segundo ciclo.

A Figura 6-31 ilustra o gráfico de comparação do desempenho *recall* para os nove ensaios no 2º ciclo. A ênfase está nos algoritmos RFC, MLP e DTC, que conseguem, até no ensaio 6, resultados com 90% de *recall*. A variação do desempenho *recall* é pequena ao longo dos nove ensaios, e mostra a estabilidade dos desempenhos obtidos em cada um dos modelos que foram avaliados. Neste gráfico, nota-se que três algoritmos RFC, MLP e DRC se isolam no topo, em comparação com os demais algoritmos, tendo um desempenho sempre próximo de 90%, o que é muito desejável e atende aos nossos critérios de aceitabilidade do modelo.

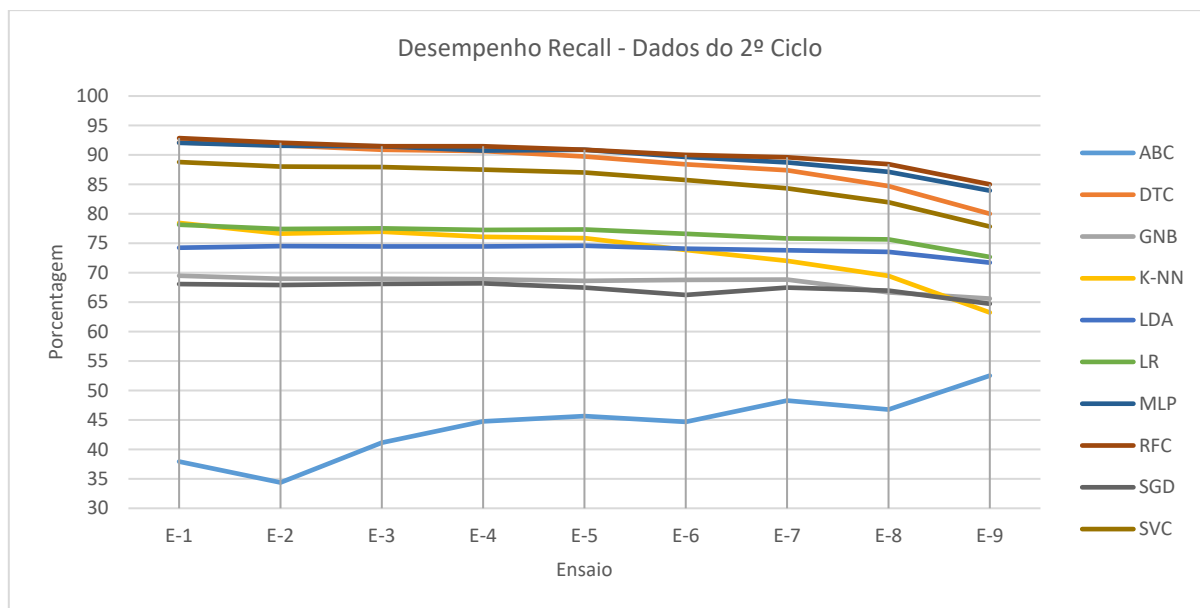


Figura 6-31. Desempenho do *recall* para os dados do segundo ciclo.

A Figura 6-32 ilustra o gráfico de comparação do desempenho *F1-Score* para os nove ensaios no segundo ciclo. Os algoritmos RFC, MLP e DTC estão novamente no topo do desempenho do *recall*. RFC e MLP são praticamente equivalentes ao longo de todos os ensaios, o que não se verifica com o DTC no E-9. O algoritmo LDA é o mais constante entre todos os algoritmos, ao longo dos nove ensaios. Isto pode ser explicado pela baixa diferença entre *precision* e *recall* deste algoritmo, uma vez que a fórmula do *F1-Score* é composta, basicamente, por essas duas medidas de desempenho.

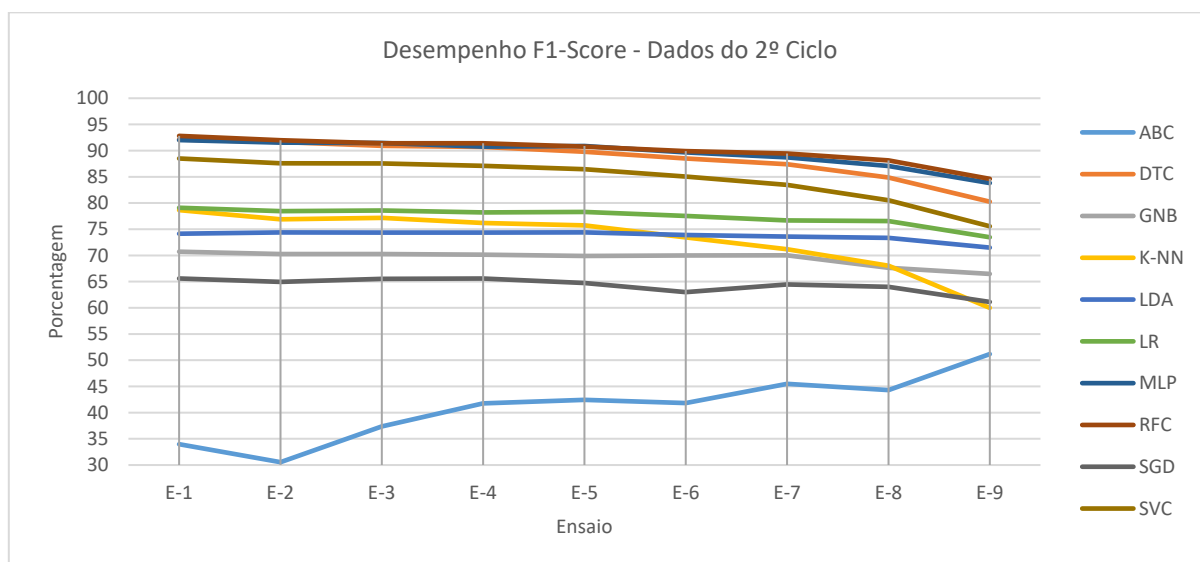


Figura 6-32. Desempenho do *F1-Score* para os dados do segundo ciclo.

Por fim, o desempenho no que diz respeito à *accuracy* é ilustrado na Figura 6-33 para os nove ensaios do 2º ciclo. Os três algoritmos RFC, MLP e DTC resultam em classificadores

com *accuracy* superior ao critério estabelecido, destacando-se o RFC, por obter modelos sempre acima de 90%, até no ensaio 6, que representa 60% dos dados de testes e 40% de treino.

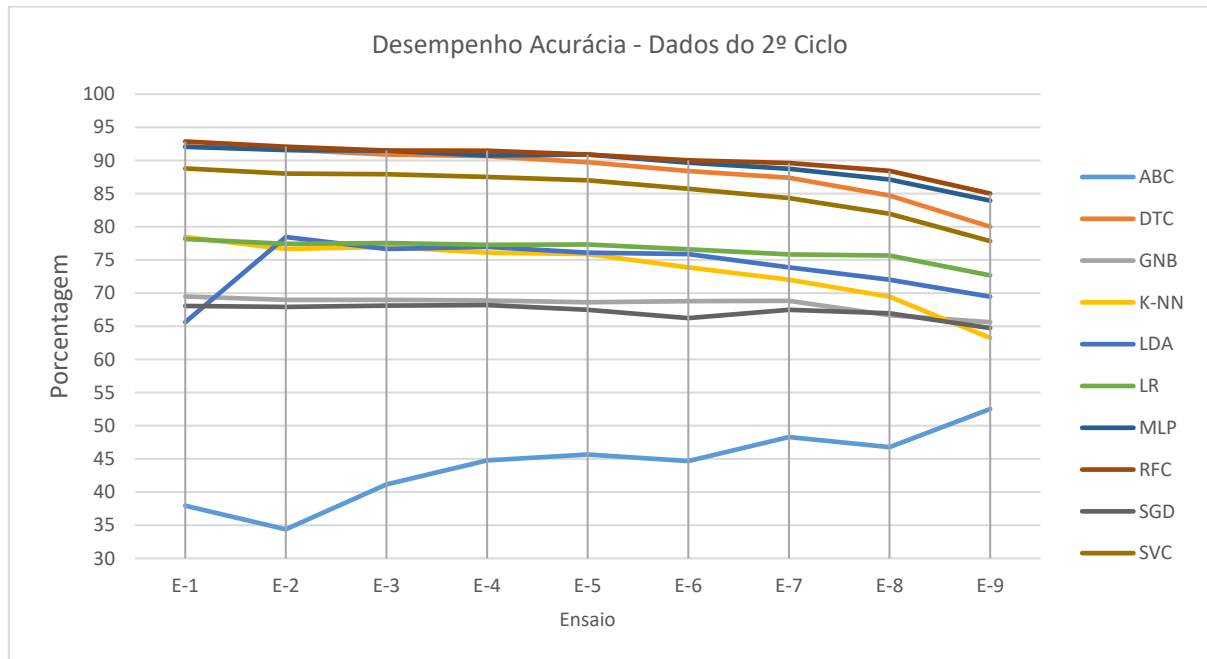


Figura 6-33. Desempenho de *accuracy* para os dados do segundo ciclo.

Os resultados obtidos dos ensaios com os 4.712 dados fornecem, de modo geral, bons modelos para os dez algoritmos utilizados, e três deles (RFC, MLP e DTC) geraram modelos acima de 90% de *precision*. Uma vez que os critérios de sucesso foram alcançados por três algoritmos, não seria necessário executar novos ciclos para investigar possíveis melhorias. Portanto, poderia ser criado um modelo cognitivo e incorporá-lo no LabDER. Entretanto, ainda não foi aplicada uma engenharia aos atributos já definidos, como um último passo importante para investigar melhorias de desempenho. Assim, definiu-se mais uma repetição no ciclo CRISP-DM, para descobrir, objetivamente, se uma engenharia aos atributos é capaz de aumentar o desempenho geral dos modelos.

6.3.3 Terceiro ciclo

6.3.3.1 Entendimento do Negócio

O objetivo do terceiro ciclo é identificar se uma mudança nos atributos pode aumentar ainda mais o desempenho dos modelos, quando aplicada ao mesmo conjunto de dados.

6.3.3.2 Compreensão dos Dados

Os dados selecionados para este ciclo são os mesmos do 2º ciclo, que contém um conjunto de 4.712 registros.

6.3.3.3 Preparação dos Dados

Neste ciclo, a partir de um conceito de teoria de conjuntos, foram desenvolvidos quatro novos atributos que representam, na sua essência, as características que estão em falta ou em excesso em relação aos tipos de atributos de cardinalidades na chave primária e chave estrangeira.

Os novos atributos que representam as informações ausentes são **NRMissingFK** (número de relacionamentos ausentes na chave estrangeira) e **NRMissingPK** (número de relacionamentos ausentes na chave primária), e dois atributos para determinar as informações que foram excedidas **NRExceedFK** (número de relacionamentos excedidos na chave estrangeira) e **NRExceedPK** (número de relacionamentos excedidos na chave primária).

Então os atributos **NRMissingFK**, **NRMissingPK**, **NRExceedFK**, e **NRExceedPK** foram gerados a partir dos atributos **NRCrdPk11**, **NRCrdPk01**, **NRCrdPk1N**, **NRCrdPk0N**, **NRCrdFk11**, **NRCrdFk01**, **NRCrdFk1N**, **NRCrdFk0N**. Deste modo, os dados foram processados de novo, e uma nova análise de correlação foi feita apenas para o novo conjunto de atributos. O resultado das correlações é apresentado na Tabela 6-9 quando comparadas com os atributos originais.

Tabela 6-9. Correlação dos novos atributos versus atributos originais.

Novos Atributos	Correlação	Atributos Originais	Correlação
NRExceedFK	-0,57	NRCrdPk01	-0,11
NRExceedPK	-0,58	NRCrdPk1N	-0,26
NRMissingFK	-0,57	NRCrdPk0N	-0,22
NRMissingPK	-0,58	NRCrdFk11	-0,31
-	-	NRCrdFk01	-0,24
-	-	NRCrdFk1N	-0,47
-	-	NRCrdFk0N	-0,25

A partir dos resultados da correlação, substituímos os atributos originais pelos atributos **NRMissingFK**, **NRMissingPK**, **NRExceedFK**, e **NRExceedPK**. No final deste processo, os seguintes atributos foram selecionados para o terceiro ciclo do estudo: **NTbl**, **NCln**, **NR**, **NRUna**, **NRBin**, **NRTm**, **NRNary**, **NR:1N**, **NR:MN**, **NREsp**, **NRFor**, **NRFra**, **NCPK**, **NCNum**, **NCCrc**, **NCDtt**, **NRMissingFK**, **NRMissingPK**, **NRExceedFK** e **NRExceedPK**.

6.3.3.4 Modelação e Avaliação

De forma análoga aos ciclos anteriores, foram obtidos mais 900 modelos com os dados de 4.712 registos do 3º ciclo, e aplicados os dez classificadores (**ABC**, **DTC**, **GNB**, **K-NN**,

LDA, LR, MLP, RFC, SGD, SVC). As informações medidas de desempenho dos modelos foram processadas (ver Anexo J: Resultado dos ensaios - 3º Ciclo), e então comparadas.

O resultado do desempenho do 3º ciclo pode ser visto na Figura 6-34, que ilustra uma comparação de desempenho (*precision*, *recall*, *F1-Score*, e *accuracy*) aplicada aos dez algoritmos anteriormente definidos (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD, SVC). Os dois algoritmos RFC e MLP são os mais eficientes, ambos com 90% de desempenho para *precision*, *recall*, *F1-Score* e *accuracy*. O terceiro algoritmo mais eficiente é o DTC, com cerca de 89% para as quatro métricas de desempenho. Seguido pelo algoritmo SVC com o quarto melhor desempenho geral. Os algoritmos ABC, GNB, e SGD têm desempenhos entre 60% a 70% nas quatro métricas.

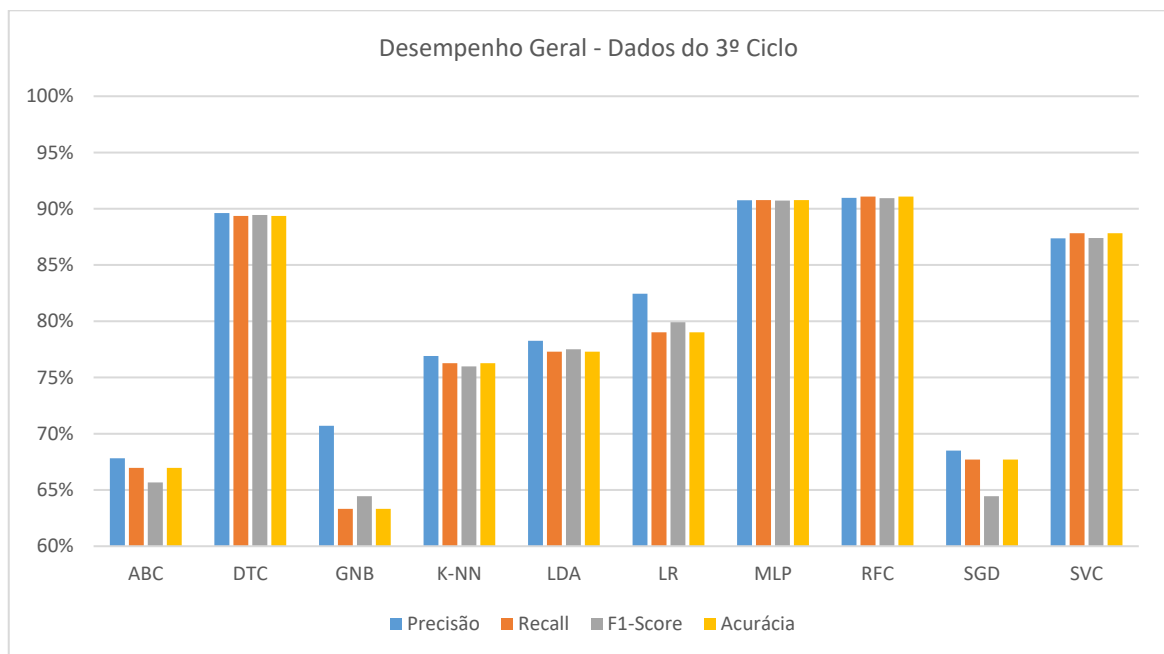


Figura 6-34. Gráfico de comparação de desempenho (*precision*, *recall*, *F1-Score*, e *accuracy*) por algoritmo de classificação (ABC, DTC, GNB, K-NN, LDA, LR, MLP, RFC, SGD e SVC) com os dados 3º ciclo.

A Tabela 6-10 apresenta uma comparação entre os ciclos do desempenho geral e as taxas de desempenho de *precision*, *recall*, *F1-Score*, e *accuracy*, com o desvio padrão para cada um dos dez classificadores anteriormente definidos. Os novos atributos aumentaram a desempenho geral de todos os modelos, ao mesmo tempo que reduziram o desvio padrão na maioria dos algoritmos. O algoritmo RFC é seguramente o mais adequado para construir um modelo cognitivo e disponibilizar a sua utilização no LabDER.

Tabela 6-10. Comparação do desempenho dos algoritmos com os dados do 1º, 2º e 3º ciclo.

Algoritmo	Ciclo	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
ABC	1º Ciclo	60,66 ±4,94	56,90 ±5,75	57,15 ±5,94	56,90 ±5,75
	2º Ciclo	50,75 ±5,72	44,02 ±5,48	40,98 ±6,22	44,02 ±5,48
	3º Ciclo	67,81 ±3,19	66,97 ±5,68	65,67 ±5,56	66,97 ±5,68
DTC	1º Ciclo	73,67 ±4,36	73,18 ±4,80	73,16 ±4,62	73,18 ±4,80
	2º Ciclo	88,66 ±3,71	88,39 ±3,91	88,47 ±3,84	88,39 ±3,91
	3º Ciclo	89,61 ±3,58	89,36 ±3,72	89,44 ±3,67	89,36 ±3,72
GNB	1º Ciclo	59,49 ±1,35	45,20 ±0,92	46,00 ±1,49	45,20 ±0,92
	2º Ciclo	73,65 ±0,81	68,31 ±1,29	69,49 ±1,43	68,31 ±1,29
	3º Ciclo	70,71 ±0,40	63,33 ±1,16	64,45 ±1,04	63,33 ±1,16
K-NN	1º Ciclo	63,65 ±4,36	64,82 ±3,58	61,97 ±4,00	64,82 ±3,58
	2º Ciclo	73,11 ±3,86	73,02 ±2,96	72,33 ±3,54	73,02 ±2,96
	3º Ciclo	76,91 ±3,78	76,27 ±3,46	75,99 ±3,76	76,27 ±3,46
LDA	1º Ciclo	66,60 ±2,09	66,59 ±2,34	66,22 ±2,14	66,59 ±2,34
	2º Ciclo	74,43 ±1,38	73,93 ±1,26	73,77 ±1,34	73,93 ±1,26
	3º Ciclo	78,26 ±1,13	77,29 ±1,19	77,50 ±1,19	77,29 ±1,19
LR	1º Ciclo	68,23 ±2,09	63,08 ±1,45	64,24 ±1,43	63,08 ±1,45
	2º Ciclo	80,36 ±1,82	76,50 ±1,65	77,43 ±1,72	76,50 ±1,65
	3º Ciclo	82,45 ±1,92	79,02 ±1,93	79,91 ±1,94	79,02 ±1,93
MLP	1º Ciclo	75,69 ±4,33	75,88 ±4,21	75,50 ±4,28	75,88 ±4,21
	2º Ciclo	89,59 ±2,69	89,57 ±2,63	89,55 ±2,66	89,57 ±2,63
	3º Ciclo	90,75 ±2,39	90,76 ±2,40	90,73 ±2,40	90,76 ±2,40
RFC	1º Ciclo	77,25 ±3,81	77,79 ±3,34	76,96 ±3,65	77,79 ±3,34
	2º Ciclo	90,07 ±2,49	90,20 ±2,37	90,06 ±2,48	90,20 ±2,37
	3º Ciclo	90,96 ±2,37	91,08 ±2,29	90,93 ±2,38	91,08 ±2,29
SGD	1º Ciclo	61,59 ±0,98	60,05 ±1,23	57,29 ±1,11	60,05 ±1,23
	2º Ciclo	67,77 ±1,48	67,24 ±1,14	64,33 ±1,47	67,24 ±1,14
	3º Ciclo	68,51 ±2,19	67,71 ±1,64	64,44 ±2,08	67,71 ±1,64
SVC	1º Ciclo	74,87 ±3,91	75,02 ±4,30	72,71 ±5,46	75,02 ±4,30
	2º Ciclo	84,45 ±3,89	85,04 ±3,58	84,17 ±4,24	85,04 ±3,58
	3º Ciclo	87,37 ±2,74	87,82 ±2,51	87,40 ±2,77	87,82 ±2,51

A comparação dos desempenhos com os dados do 1º, 2º e 3º ciclos é ilustrada no gráfico da Figura 6-35. Durante os três ciclos de execução, existe um aumento linear nos desempenhos, com exceção de dois algoritmos: o ABC que, ao invés de aumentar, diminuiu no 2º ciclo, e o GNB que, ao invés de aumentar, diminuiu no 3º ciclo. Com exceção desses dois algoritmos, todos os outros oito aumentaram de forma linear durante todos os ciclos. No geral, houve algoritmos que aumentaram o seu desempenho cerca de 5% ou mais, em relação ao ciclo anterior, e cerca de 1% nos algoritmos que parecem chegar ao limite de aprendizagem com os dados existentes.

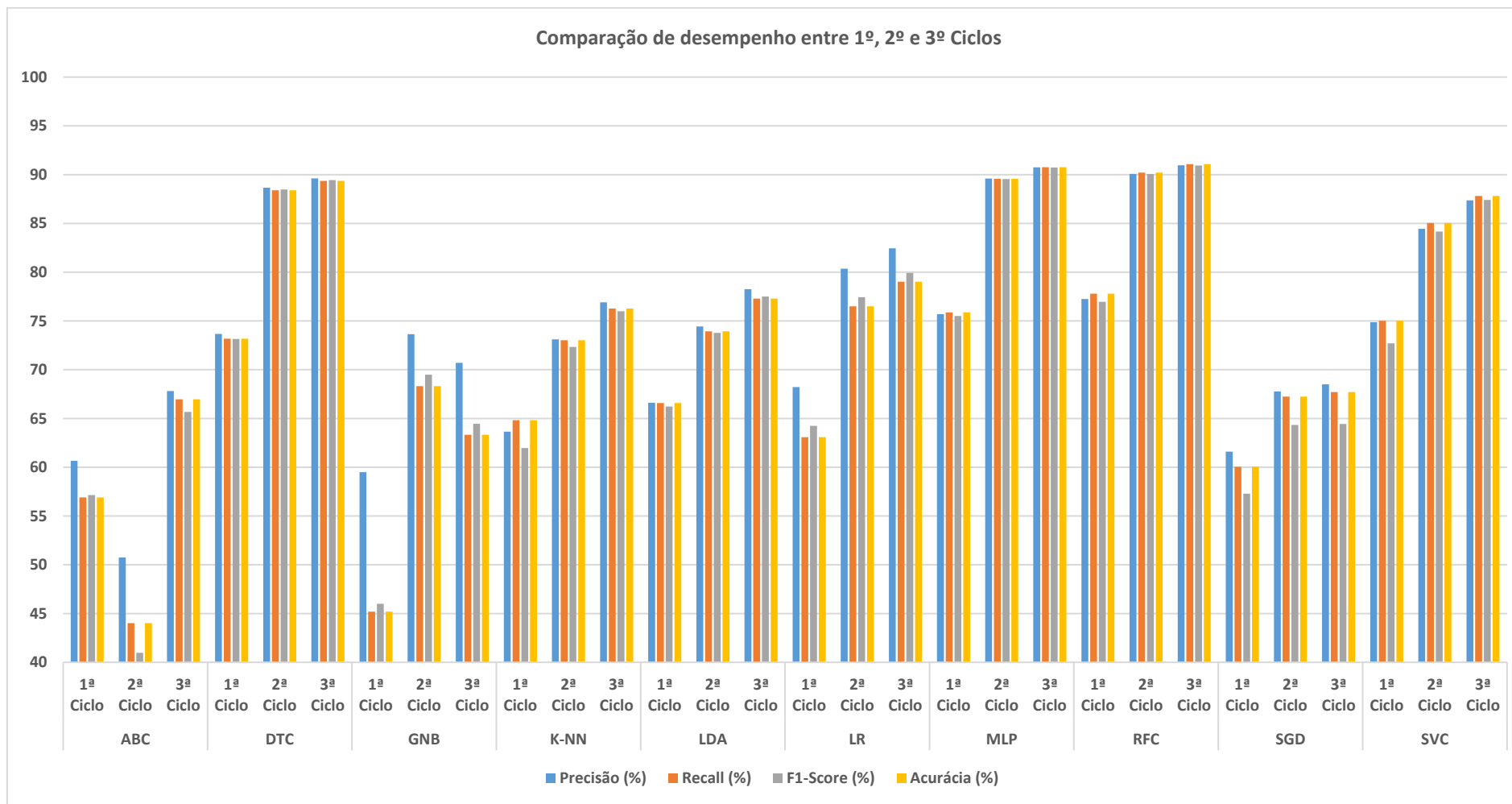


Figura 6-35. Gráfico comparativo de desempenho entre os 1º, 2º, e 3º ciclos.

A Figura 6-36 ilustra o gráfico do desempenho de *precision* dos classificadores, em cada um dos nove ensaios do 3º ciclo. Nota-se uma vantagem elevada de *precision* quando comparada com o ciclo anterior. Os três algoritmos, que se isolaram no topo, estão realmente acima dos 90%, mesmo com poucos dados de treino, como no ensaio 6. Os novos atributos desenvolvidos são, de longe, superiores aos do 2º ciclo, sendo muito úteis para os algoritmos aprenderem com os dados. Isso pode ser observado pela pequena oscilação do algoritmo ABC, que era sempre muito elevada nos ciclos anteriores e, que neste gráfico (ver Figura 6-36), se apresenta mais uniforme, em relação aos ciclos anteriores.

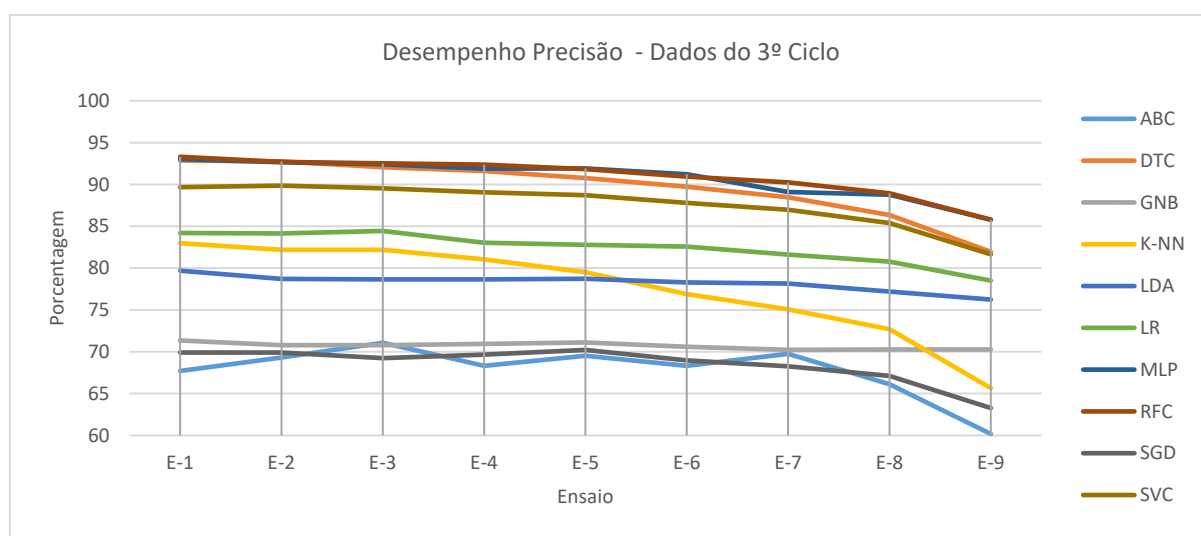


Figura 6-36. Desempenho de *precision* para os dados do 3º ciclo.

A Figura 6-37 ilustra o gráfico de comparação do desempenho *recall* para os nove ensaios no 3º ciclo. Os algoritmos RFC, MLP e DTC continuam à frente dos demais algoritmos, e sempre acima dos 90%, inclusive no sétimo ensaio. O algoritmo ABC supera os algoritmos GNB e SGD em alguns ensaios, o que mostra novamente o efeito positivo dos novos atributos.

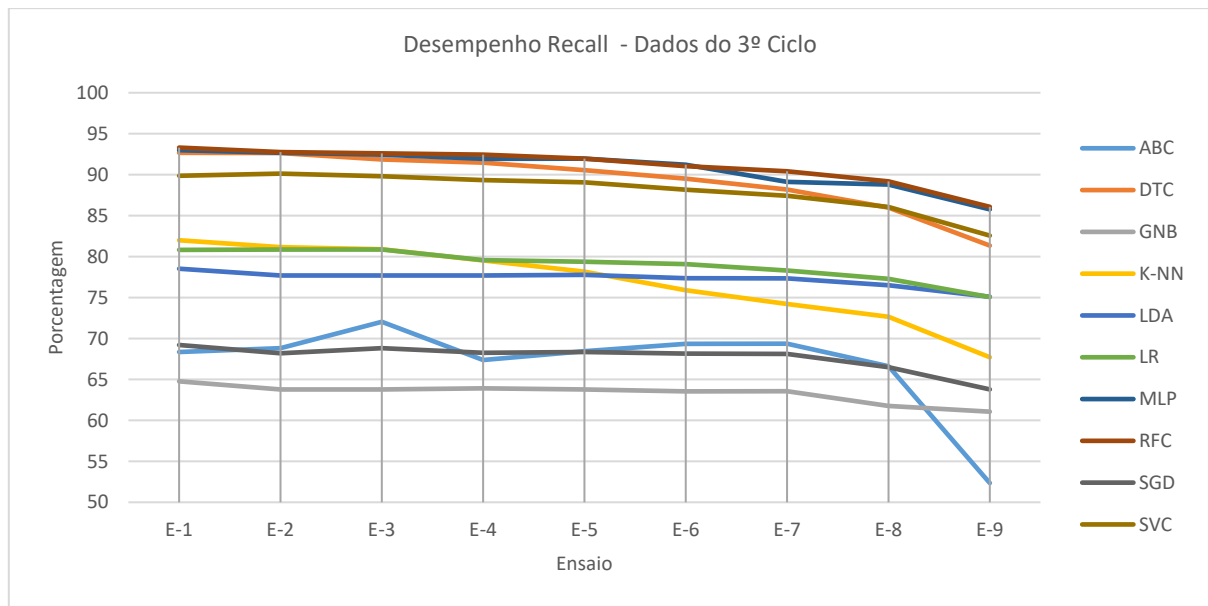


Figura 6-37. Desempenho do *recall* para os dados do 3º ciclo.

A Figura 6-38 ilustra o gráfico de comparação do desempenho *F1-Score* para os nove ensaios no 3º ciclo. O aumento global do *F1-Score* também se nota neste gráfico. A maioria dos gráficos de desempenho do ciclo anterior estavam, no início da escala, nos 30%, o que se devia ao algoritmo ABC. Se não se considerar o ensaio com 10% de dados de treino, pode afirmar-se que o aumento é de 30% para o algoritmo ABC, com os novos atributos. A maioria dos algoritmos tiveram um aumento de *F1-Score*, o que significa que tanto o *recall* como a *precision* estão mais equilibrados.

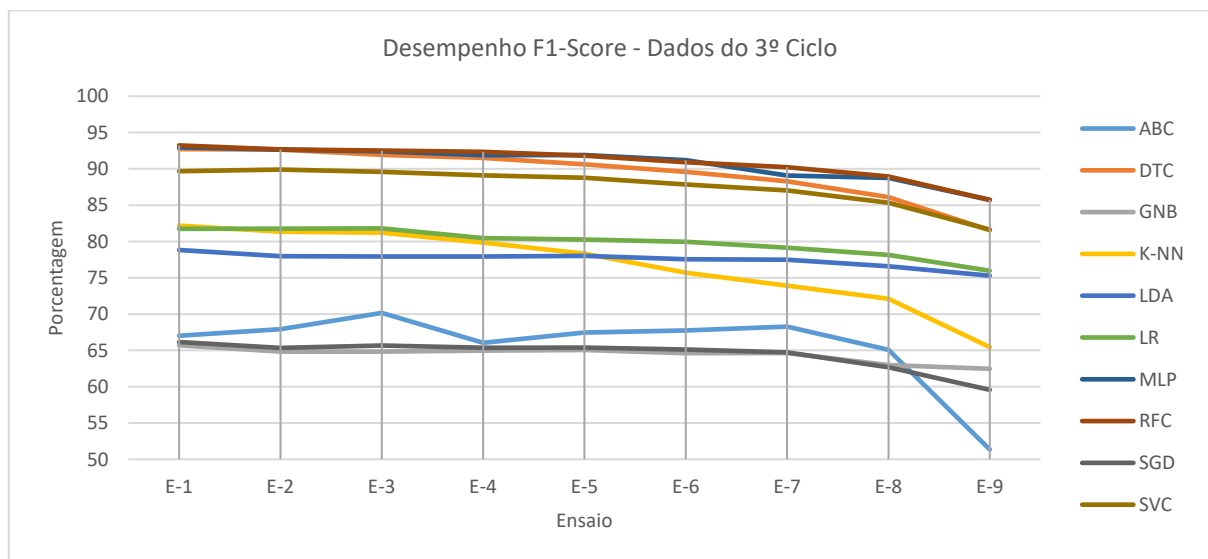


Figura 6-38. Desempenho do *F1-Score* para os dados do 3º ciclo.

O desempenho de *accuracy* é ilustrado na Figura 6-39 para os nove ensaios do 3º ciclo. Os três algoritmos RFC, MLP, e DTC têm desempenhos acima dos 90% ao longo dos ensaios. Os novos atributos deram características importantes ao algoritmo SGD, neste ciclo, que chegou a aumentar a *accuracy* em cerca de 10%, em relação ao algoritmo anterior. No geral, a reestruturação dos atributos ajudou na construção de modelos mais eficientes.

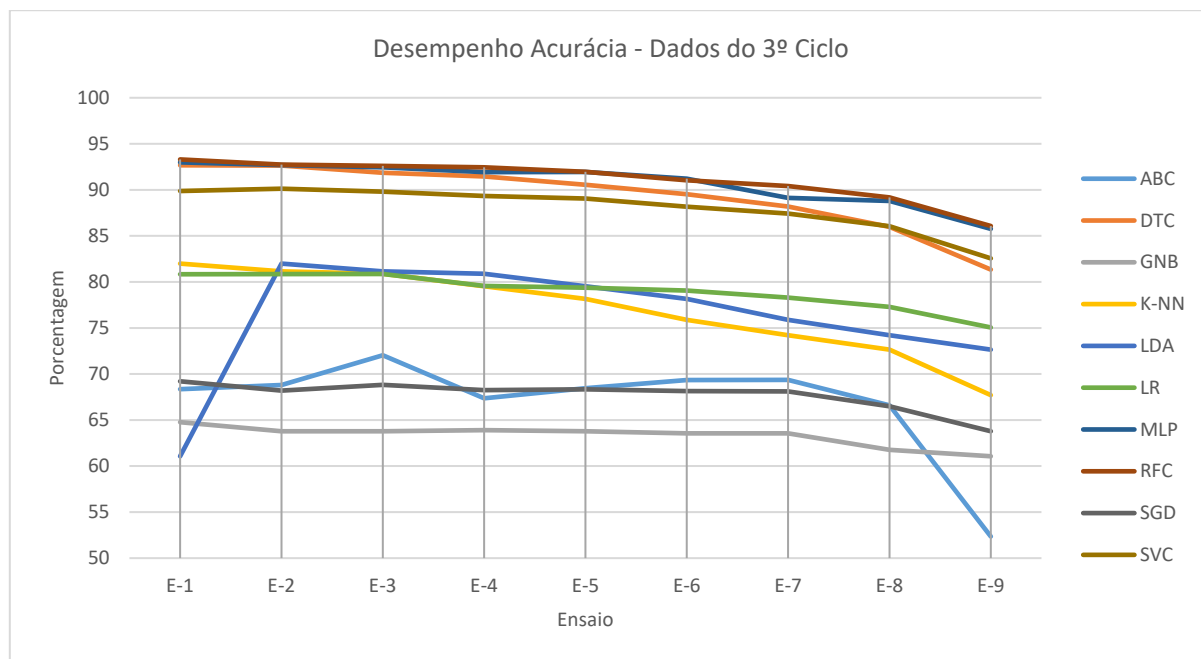


Figura 6-39. Desempenho de *accuracy* para os dados do 3º ciclo.

Foi apresentada uma série de evidências de aumento do desempenho, a partir da aplicação da reestruturação dos atributos, no conjunto de 4.712 dados. Pode parecer irrelevante o aumento de 1% nos índices de desempenho dos algoritmos RFC, MLP e DTC, mas isso pode estar associado ao contexto de aplicação do estudo ou às formas de interpretar esse ganho percentual. Se o contexto fosse na área da saúde, segurança ou na área financeira, provavelmente esse 1% seria visto como um resultado muito positivo, devido ao apelo da área de aplicação. No contexto da educação, esse 1% tem mais impacto quando associado às avaliações do professor, pois representa o quão próximo a avaliação automática está de um avaliador humano. Ao mesmo tempo, evita, ao máximo, que o sistema cometa erros de classificação. Por isso o *recall* é tão importante como a *accuracy*.

É importante lembrar o propósito do uso da aprendizagem computacional e o seu papel no *framework* de detecção automática. O principal objetivo era encontrar um modelo que fosse capaz de avaliar, automaticamente, um diagrama ER e tivesse uma taxa de sucesso próxima da de um avaliador humano. Desta forma, foi determinado o critério mínimo de 90% e

desenvolvido todo o planejamento para atender aos requisitos de alto nível de exigência e, conseqüentemente, colocar em produção algo que fosse útil para professores e alunos.

Um outro detalhe importante é o de que ainda existem oportunidades de melhoria a serem exploradas, pois apenas uma parte dos atributos foi utilizada no processo de reestruturação. Também há a hipótese de melhorias a partir de outras configurações dos algoritmos utilizados, pois foi aplicada apenas uma configuração padrão dos algoritmos de aprendizagem computacional disponíveis na biblioteca scikit-learn (Pedregosa et al., 2011). Essas oportunidades de melhoria podem ser exploradas futuramente.

6.4 Avaliação de campo do LabDER

Este ensaio visou validar a aplicação da avaliação automática em sala de aula, com ênfase na avaliação de DER. Para isso, o LabDER foi aplicado como ambiente de aprendizagem sobre bases de dados relacionais durante um semestre letivo na instituição de ensino ISCAC, onde alunos e professores manipularam o LabDER e puderam testar todos os recursos disponíveis. Uma série de evidências foram recolhidas, durante seis meses, de forma a atender aos objetivos da avaliação do LabDER. Os objetivos desta avaliação foram:

(1) identificar a participação do aluno: a participação do aluno é determinada pelo número de respostas, por tipo de questão e acessos, durante o curso;

(2) medir o desempenho dos alunos na disciplina: o desempenho dos alunos é medido pela frequência e percentual de acerto, de cada conceito de DER submetido;

(3) identificar se o modelo da avaliação automática de DER pode ser utilizado satisfatoriamente em sala de aula: a avaliação do modelo é medida a partir das respostas ao inquérito submetido aos professores.

6.4.1 População

O estudo do caso foi realizado no curso de Licenciatura em Informática de Gestão do ISCAC. De entre as seis universidades onde o LabDER foi implantado, o ensaio no ISCAC foi conduzido presencialmente, com reuniões e suporte em aulas presenciais, e teve a avaliação automática com o *feedback* semântico do DER, aplicado na sua íntegra.

As aulas decorreram, presencialmente, em duas turmas, uma no período diurno (19 alunos) e outra no período noturno (8 alunos), durante 15 semanas letivas, entre os meses de fevereiro a julho de 2019, totalizando uma carga horária de 67.5 horas de curso. Os alunos não

tinham conhecimentos prévios de bases de dados, mas tiveram contacto com as disciplinas de Sistemas Operativos e Linguagens de Programação.

O ISCAC pôs ao dispor dos professores e alunos uma infraestrutura com espaços de convívio comum com acesso à internet e diversos laboratórios informáticos equipados com computadores desktops/ de mesa, onde a maioria das aulas decorreram com o uso do ambiente LabDER (ver Figura 6-40).



Figura 6-40. ISCAC - Área comum de estudo com acesso à internet.

O laboratório esteve sempre na sua capacidade máxima de alunos. No dia de exames formais, as turmas da noite e da manhã juntavam-se e, portanto, os alunos eram divididos entre dois laboratórios. A Figura 6-41 ilustra os alunos a utilizar o LabDER na sua última avaliação que decorreu a 11 de julho de 2019.



Figura 6-41. Última avaliação formal aplicada no ISCAC, a 11 de julho de 2019.

A seção a seguir descreve a metodologia e instrumentos utilizados na avaliação do LabDER.

6.4.2 Metodologia e instrumentos

A primeira atividade visou preparar o ambiente de trabalho e depois formar os professores na utilização dos recursos do LabDER, permitindo que executassem o plano de ensino. A formação teve o plano de ensino do professor como guia, que foi convertido num conjunto de atividades executáveis no LabDER (ver Tabela 6-11). Deste modo, os dados foram adicionados ao ambiente de trabalho, como curso, disciplina, questões, respostas e atividades.

Os alunos receberam formação em LabDER no primeiro dia de aula, imediatamente após a explicação das atividades previstas no plano de ensino do professor (consulte a Tabela 6-11) que detalha a composição da nota final do aluno, conforme segue:

$$CF = (A1 * 10\% + A2 * 10\% + A3 * 10\% + F1 * 35\% + F2 * 15\% + F3 * 20\%).$$

A formação em LabDER visou garantir o primeiro acesso de todos os alunos e explicar as funcionalidades do ambiente de trabalho, incluindo a avaliação automática do sistema. Na

formação, também foi explicado o papel do investigador como o encarregado pela proteção de dados do site conforme o RGPD, e foram feitos esclarecimentos sobre a plataforma.

Tabela 6-11. Plano de atividades do curso por período e conteúdo. SQL é denominado como (S), DER é denominado (E), questões de texto livres são denominadas (D) e questões de escolha múltipla são denominadas (M).

Ati.	Questões	Período DD/MM	Capítulos do Livro LabDER	Conteúdo	
A1	S D M	20 18 23	18/02 a 25/03	Módulo I	Conceitos de bases de dados, SGBD, Álgebra relacional, visão geral de SQL.
				Módulo II	Elementos de sintaxe SQL, criação de sintaxe BD, consultas, restrições, classificação e operadores lógicos e aritméticos.
				Módulo III	Instruções DDL e DCL, Criar / Excluir Objetos, Instruções Insert/Update/Delete, Sintaxe de Restrições, Índices, Visões, Procedimentos, Gatilhos.
				Módulo IV	Instruções SQL GROUP BY, Consultas complexas com HAVING.
A2	S D M	17 06 28	27/03 a 15/04	Módulo V	Tipos de junções: cross, inner, outer, left, right. Subconsultas: na Coluna como uma expressão escalar, na cláusula WHERE como um Filtro, na cláusula FROM como uma Tabela, na cláusula HAVING como uma Seleção de Grupo, Consultas complexas misturando todos os módulos SQL.
A3	E M	25 35	17/04 a 10/06	Módulo VI	Conceitos de design: modelo de bases de dados relacionais, modelo de entidade e relação (ER), modelação de dados avançada, normalização de bases de dados.
E1	S M	07 03	26/03	Módulo I to, II, III, IV	Exame 1 - Avalie todos os conceitos introdutórios de bases de dados e SQL.
E2	S M	07 03	16/04	Módulo V	Exame 2 - Avalie os conceitos avançados de SQL.
E3	E M	04 08	11/06	Módulo VI	Exame 3 - Avalie todos os conceitos de DER.

Durante os primeiros dias de aulas, foi dado suporte presencial nas aulas na forma de exercícios práticos. Ao longo do semestre letivo, foram realizados outros suportes presenciais a pedido dos professores e alunos, como por exemplo, em dias que antecederiam os exames.

Com a ajuda do livro digital e pequenos exemplos do mundo real, foram ensinados aos alunos conceitos de BD relacionais, SQL e modelo ER ao longo de cinco semanas. Os conceitos lecionados estão listados na Tabela 6-11 (módulos I a IV). Porém, tiveram a primeira atividade (A1) com 61 questões (ver Tabela 6-11). No final do período permitido para esta atividade, os alunos tiveram um exame (E1) com 10 questões.

Nas três semanas seguintes, os conceitos de junções e subconsultas foram ensinados aos alunos, conforme descrito detalhadamente na Tabela 6-11 (módulo V). Os alunos tiveram a atividade 2 (A2) para conclusão, no final deste período de três semanas. A A2 misturou os conceitos de junções e subconsultas com funções agregadas e tópicos de módulos anteriores e teve 51 questões. No final da A2, os alunos tiveram outro exame (E2) com 10 questões.

Nas sete semanas seguintes, foram ensinados aos alunos os conceitos de modelação de bases de dados descritos na Tabela 6-11 (módulo VI). Os alunos tiveram a atividade 3 (A3), com 60 questões, a serem submetidas no final deste período de sete semanas. Os alunos tiveram o último exame (E3) após a A3. Este exame tem 11 questões.

Estes seis conjuntos de dados (A1, A2, A3, E1, E2 e E3) foram recolhidos (ao longo de um período de 15 semanas, no total) e avaliados automaticamente pelo LabDER. Os logs dos dados foram então analisados para identificar a frequência e percentual de uso das funcionalidades do sistema, respondendo os objetivos (1) e (2) dessa avaliação. Um questionário foi aplicado aos professores para responder à dimensão (3) da avaliação do ambiente de trabalho.

6.4.3 Análise de resultados

6.4.3.1 Resultados gerais do uso do LabDER

Os resultados gerais apresentam as informações mais relevantes que comprovam a participação do aluno, o número de respostas submetidas por tipo de questão, número de acessos e outros log do sistema. Com estas informações, esta secção visa responder à dimensão **(1) identificar a participação do aluno.**

Adicionalmente, os componentes do DER do aluno e do DER do professor eram identificados de acordo com as métricas estabelecidas. Desta forma, foi identificada a frequência geral de cada componente de DER do aluno, por avaliação, e a percentagem de acertos do aluno. Esses dados estão descritos na Tabela 6-13 que, em resumo, responde à dimensão **(2) medir o desempenho dos alunos na disciplina.**

Um total de 15.158 respostas foram recolhidas no contexto das avaliações. Destas, 2.539 foram texto livre, 2.435 foram respostas de escolha múltipla, 6.097 foram respostas SQL e 4.087 foram respostas DER (ver Tabela 6-12). O log do sistema registou uma frequência de 1.915 logins e 26.918 interações no sistema. As interações incluem visualização de livros digitais, exercícios, relatórios e exibição de janela do sistema. No SQL-Livre, mais de 13.000 comandos foram submetidos, e cerca de 800 diagramas, utilizando o DER-Livre.

Tabela 6-12. Número de respostas enviadas por tipo de questão por log geral do sistema.

<i>Atividade (Total de questões)</i>	<i>Respostas enviadas</i>					<i>Log do utilizador</i>			
	<i>De texto livre</i>	<i>Escolha múltipla</i>	<i>SQL</i>	<i>DER</i>	<i>Total</i>	<i>Login</i>	<i>Iterações</i>	<i>SQL Livre</i>	<i>DER Livre</i>
<i>Atividade 1 (61 q)</i>	2.106	551	2.269	N/A	4.926	589	9.698	5.144	N/A
<i>Atividade 2 (51 q)</i>	433	701	2.028	N/A	3.162	486	8.165	5.149	N/A
<i>Atividade 3 (60 q)</i>	N/A	815	N/A	3.197	4.012	743	7.027	830	809
<i>Exame 1 (10)</i>	N/A	87	832	N/A	919	30	819	992	N/A
<i>Exame 2 (10)</i>	N/A	113	968	N/A	1.081	40	967	1.680	N/A
<i>Exame 3 (11)</i>	N/A	168	N/A	890	1.058	27	242	N/A	N/A
Total	2.539	2.435	6.097	4.087	15.158	1.915	26.918	13.795	809

No período de fevereiro a julho de 2019, o log do Google Analytics aponta o acesso do LabDER com origem em diversas localidades e regiões de Portugal Continental e Insular (ver

Figura 6-42). Os acessos foram confirmados pelos alunos como tendo sido feitos a partir das residências de seus familiares.

A participação do aluno, no ISCAC, ocorreu também muitas vezes fora da instituição de ensino, e era normalmente feito o acesso a partir das residências de familiares. A Figura 6-42 ilustra o acesso dos alunos do ISCAC ao LabDER, com origem em diferentes regiões de Portugal, durante o período do curso.

Destaca-se o número de acesso devido à autonomia fornecida ao aluno, que realizou o acesso ao ambiente de trabalho, a partir de distintas localidades, de forma a continuar o processo de aprendizagem proporcionada pelo *feedback* semântico e pela avaliação automática. Na publicação mais recente da ferramenta AsseSQL, em 9 anos, cerca de 2.300 alunos enviaram 161 mil respostas de SQL nas avaliações (Ahadi, Behbood, Vihavainen, Prior, & Lister, 2016). Ou seja, uma média geral de 70 respostas SQL por aluno. No LabDER, a média de respostas de SQL por aluno, no ISCAC, foi de 265 respostas, enquanto de DER a média geral foi de 177 respostas por aluno, considerando o total de submissões SQL e de DER descritas na Tabela 6-12 e o total 23 alunos.

A média geral expressa indica que os alunos exploram o LabDER quase quatro vezes mais do que ambientes de trabalho da mesma categoria. Desconsideram-se, neste cálculo, as submissões de treinos desses alunos, sendo que muitos deles preferem usar a interface de treino para submeter apenas as respostas corretas registadas no histórico de respostas. Neste caso, se fossem incluídos os dados de treino de SQL, a média geral seria de 864 respostas por aluno. Existem mais de 500 mil respostas submetidas desde a primeira versão do sistema, e a média anual é sempre superior aos demais ambientes de trabalho (A. Lino et al., 2019). Esta evidência revela o esforço que o aluno exerce para aprender e tentar sempre melhorar a nota final.

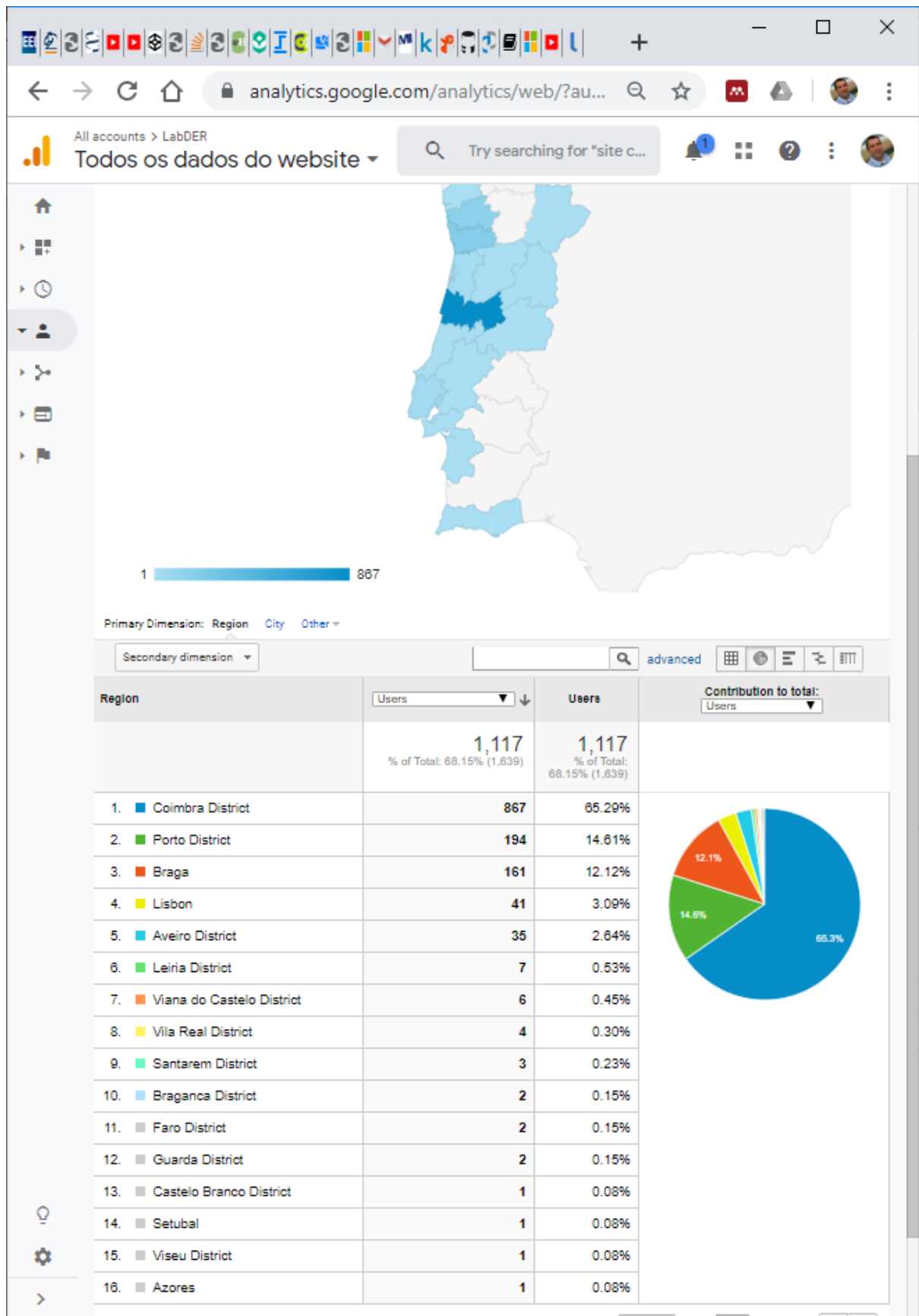


Figura 6-42. Utilizadores do ISCAC em Portugal durante os meses de fevereiro a julho de 2019.

6.4.3.2 Resultados - avaliação DER

Em cada resposta submetida, os componentes do DER do aluno e do DER do professor, eram identificados de acordo com as métricas estabelecidas. Desta forma, foi identificada a frequência geral de cada componente de DER do aluno por avaliação, e a percentagem de acertos do aluno. Estes dados estão descritos na Tabela 6-13 que, em resumo, responde à dimensão (2) **medir o desempenho dos alunos na disciplina**.

A Tabela 6-13 apresenta os conceitos de DER avaliados no ISCAC nas atividades A3 e E3, contendo a métrica utilizada do cálculo, a frequência e o percentual de acerto por componente de DER. Por exemplo, na atividade 3 tem 25 questões de DER (ver Tabela 6-13), e teve uma frequência de 23.764 colunas do tipo numérico, sendo que 81% desta frequência está correta.

Tabela 6-13. Frequência e percentagem de acerto dos componentes do DER avaliado no ISCAC, por métricas e atividades.

Conceitos de DER Avaliados no ISCAC	Métricas Utilizadas	Atividade 3		Exame 3	
		Frequência	Acertos	Frequência	Acertos
Tabelas	NT	12 613	89%	2 660	76%
Colunas	NC	43 624	76%	7 998	69%
Domínio	NCNum	23 764	81%	3 529	73%
Domínio	NCCrc	16 212	77%	3 850	76%
Domínio	NCDtt	3 644	96%	615	89%
Chave Primária	NCPk	14 899	83%	3 132	78%
Relacionamento	NR	10 214	86%	2 089	84%
Relacionamento	NR:1N	6 788	78%	1 030	70%
Relacionamento	NR:MN	1 655	86%	233	75%
Grau do relacionamento	NRUna	115	98%	3	99%
Grau do relacionamento	NRBin	2 337	88%	413	83%
Grau do relacionamento	NRTer	300	99%	107	95%
Grau do relacionamento	NRNary	273	99%	44	99%
Especialização/generalização	NREsp	926	92%	238	93%
Força do relacionamento	NRFor	3 798	86%	391	89%
Força do relacionamento	NRFra	6 416	83%	1 698	82%
Cardinalidade/ Participação	NRCrdPk11	7 068	76%	1 101	71%
Cardinalidade/ Participação	NRCrdPk01	729	91%	169	99%
Cardinalidade/ Participação	NRCrdPk1N	998	90%	466	80%
Cardinalidade/ Participação	NRCrdPk0N	738	97%	56	99%
Cardinalidade/ Participação	NRCrdFk11	1 559	89%	580	78%
Cardinalidade/ Participação	NRCrdFk01	527	92%	108	93%
Cardinalidade/ Participação	NRCrdFk1N	5 111	74%	979	71%
Cardinalidade/ Participação	NRCrdFk0N	2 330	79%	132	95%

A principal vantagem na aplicação da abordagem de avaliação automática é o mapeamento entre os conceitos de DER, em formas de métricas, como pode ser visto na Tabela 6-13. Esta abordagem permitiu identificar exatamente a frequência de cada um dos componentes de DER, e a percentagem de acerto que os alunos conseguiram. Esta informação obviamente é por aluno, mas aqui é sumarizada por atividade.

A Figura 6-43 ilustra a resposta do aluno à questão 273, e o *feedback* automático em vermelho que fornece informações sobre a falta de um relacionamento M:N e o número de cardinalidades erradas na origem e destino do relacionamento. Adicionalmente, o sistema atribuiu automaticamente a nota 2 ao DER do aluno, e o professor adicionou o *feedback* manual em amarelo: “A relação entre o Benefício e o Plano está invertida. Ao corrigir isso, também solucionará o erro da relação M:N que está em falta.”.

A interface do aluno é semelhante à do professor, mas um botão de envio de resposta é visível em vez do botão de alteração de nota. A resposta de cada aluno é salva e avaliada automaticamente, e os resultados são apresentados no painel de *feedback* semântico e no painel histórico. O aluno pode ver as avaliações manuais do professor através do painel Histórico, no formato *post-it*.

LabDER - Laboratório de Ensino e Aprendizagem de DER - Exercícios - Mozilla Firefox

https://labder.dei.uc.pt/sql_exercicio.php?cdProb=273&cdResposta=3&idaval=325&cdConteudo=80&cdUsrAluno=10918616539&situa...

Questão 268 269 270 271 272 273 274 275 276 277 279 280 281 285 288 289 291 294 295 299 300 301 303 304 307

Atividade 3 - Parte A (Módulo VI) Início:13/05/2019 07:00:00 - Término:06/06/2019 23:59:59

273-Crie o diagrama relacional para mostrar as relações entre EMPREGADO, BENEFÍCIO, TRABALHO e PLANO. Use o banco de dados mostrado na Figura P3.10.

Observe que o banco de dados é composto de quatro tabelas que refletem esses relacionamentos:

- Um EMPREGADO tem apenas um JOB_CODE, mas um JOB_CODE pode ser mantido por muitos EMPREGADOS.
- Um EMPREGADO pode participar de muitos PLANOS, e qualquer PLANO pode ser atribuído a muitos EMPREGADOS.

Observe também que o relacionamento M: N foi dividido em dois relacionamentos 1: M para os quais a tabela BENEFIT atua como a entidade composta ou ponte.

O relacionamento entre Beneficio e Plano está invertido. Ao corrigir isso, também solucionará o erro da relação MN que está faltando.

Created by Professor

Alterar Nota Nota=2 (21% - 49%) - Fraco Máx.Pontos:5.00

Feedback Semântico

Está faltando na resposta: Relacionamento (M:N).

Dica de Cardinalidade

Quantidade de cardinalidades erradas na origem:1

Quantidade de cardinalidades erradas no destino:1

Pontuação sugerida:2

Q.268	Q.269	Q.270	Q.271	Q.272	Q.273	Q.274	Q.275	Q.276	Q.277
4.00	1.00	2.00	1.00	4.00	2.00	1.00	1.00	3.00	2.00
4.00	5.00	5.00	5.00		2.00	1.00	1.00	3.00	2.00
5.00	5.00	5.00	5.00		5.00	2.00	2.00	3.00	2.00
			5.00			5.00	2.00	3.00	2.00
						5.00	2.00	3.00	2.00
						5.00	4.00	3.00	2.00

LabDER - Laboratório de Ensino e Aprendizagem de DER - Sc...

https://labder.dei.uc.pt/sql_schemabd.php

Table name: EMPLOYEE

EMP_CODE	EMP_LNAME	JOB_CODE
14	Rudell	2
15	McDade	1
16	Ruellardo	1
17	Smith	3
20	Smith	2

Table name: BENEFIT

EMP_CODE	PLAN_CODE
15	2
15	3
16	1
17	1
17	3
17	4
20	3

Table name: JOB

JOB_CODE	JOB_DESCRIPTION
1	Clerical
2	Technical
3	Managerial

Table name: PLAN

PLAN_CODE	PLAN_DESCRIPTION
1	Term life
2	Stock purchase
3	Long-term disability
4	Dental

fechar

Figura 6-43. Interface do professor que avalia o DER do aluno com *feedback* automática e manual - Tipo de questão Transcrição.

Outro exemplo, é ilustrado pela Figura 6-44 que mostra a interface do professor com a resposta do aluno à questão 304. O sistema avaliou como 1 a nota do DER e emitiu *feedback* automático em vermelho, que fornece as informações “excedeu o máximo de” (colunas do domínio caráter, chave primária, e relacionamento forte.) e “está em falta na resposta” (colunas do domínio numérico, relacionamento 1:N, relacionamento M:N, e relacionamento fraco). Este painel também informa qual é a quantidade de cardinalidade errada no relacionamento, neste caso, uma na origem e uma no destino, assim como a nota de classificação final.

LabDER - Laboratório de Ensino e Aprendizagem de DER - Exercícios - Mozilla Firefox

https://labder.dei.uc.pt/sql_exercicio.php?cdProb=275&cdResposta=288&idaval=325&cdConteudo=80&cdUsrAluno=10918616534&situacaoProb=A

Questão 268 269 270 271 272 273 274 275 276 277 279 280 281 285 288 289 291 294 295 299 300 301 303 304 307

Atividade 3 - Parte A (Módulo VI) Início:13/05/2019 07:00:00 - Término:06/06/2019 23:59:59

304-Crie um diagrama relacional sobre a loja e seus funcionários que funcionem em diferentes regiões. Observe, em particular, como as regras de negociação refletem nas tabelas e seu conteúdo:

- Uma região pode ser o local de muitas lojas.
- Cada loja está localizada em apenas uma região.
- Cada loja emprega um ou mais funcionários.
- Cada empregado é empregado por uma loja. (Nesse caso, estamos assumindo que a regra de negócios específica que um funcionário não pode trabalhar em mais de uma loja por vez.)
- Um trabalho - como contador ou representante de vendas - pode ser atribuído a muitos funcionários. (Por exemplo, seria razoável supor que uma loja pode ter mais de um representante de vendas.
- Portanto, o cargo (Representante de Vendas) pode ser atribuído a mais de um funcionário por vez.
- Cada funcionário pode ter apenas uma atribuição de trabalho. (Nesse caso, estamos assumindo que a regra de negócios específica que um funcionário não pode ter mais de uma atribuição de trabalho de cada vez.)
- Cada região tem uma descrição, que permite identificar a região da loja. Por exemplo, Coimbra shopping, Castelo de São Jorge, cai do Iguazu, Copacabana.
- Todo empregado tem um nome, endereço, sexo e data de nascimento.
- Cada trabalho tem um acrônimo, descrição e um salário base.

Alterar Nota Nota=1 (0% - 20%) - Falhou Máx.Pontos:5.00

Feedback Semântico

Excedeu o máximo de: Colunas do domínio caracter. Primary Key. Relacionamento forte (PK,FK).

Está faltando na resposta: Colunas do domínio numérico. Relacionamento (1:1). Relacionamento (M:N). Relacionamento fraco (FK).

Dica de Cardinalidade

Quantidade de cardinalidades erradas na origem:1
Quantidade de cardinalidades erradas no destino:3
Pontuação sugerida:1

Q.268	Q.269	Q.270	Q.271	Q.272	Q.273	Q.274	Q.275	Q.276	Q.277	Q.279	Q.
5.00	1.00	3.00	2.00	2.00	5.00	2.00	1.00	3.00	4.00	1.00	3.00
5.00	1.00	3.00	2.00	3.00	5.00	1.00	2.00	5.00	4.00	4.00	5.00
2.00	3.00	3.00	3.00	3.00	5.00	1.00	4.00	5.00	4.00	4.00	4.00
4.00	4.00	3.00	5.00	3.00	3.00	4.00	4.00	4.00	4.00	4.00	4.00
3.00	4.00	4.00	3.00	3.00	3.00	5.00	3.00	5.00	4.00	4.00	4.00
1.00	4.00	4.00	3.00	3.00	3.00	3.00	4.00	4.00	4.00	4.00	4.00

Figura 6-44. Interface do professor que avalia o DER do aluno com *feedback* automático - Tipo de questão Cenário.

6.4.3.3 Resultados - avaliação SQL

O resultado a ser analisado é a resposta do aluno à questão 170. A Figura 6-45 mostra a interface do professor ao avaliar o comando SQL de um aluno. No painel de comentários, o resultado da avaliação automática pode ser visto, classificando o SQL como 92,43% correto e fornecendo *feedback* semântico com dicas de aperfeiçoamento de comando SQL.

A solução do aluno para a questão 170, quando comparada com a resposta do sistema, requer mais uma expressão de comparação para produzir o mesmo resultado SQL. Portanto, a dica de *feedback* automático notifica o aluno que o SQL excedeu o número de operadores de comparações e indica os operadores excedidos entre parênteses. Neste caso, o sinal de igualdade e dos operadores “select, from, in”.

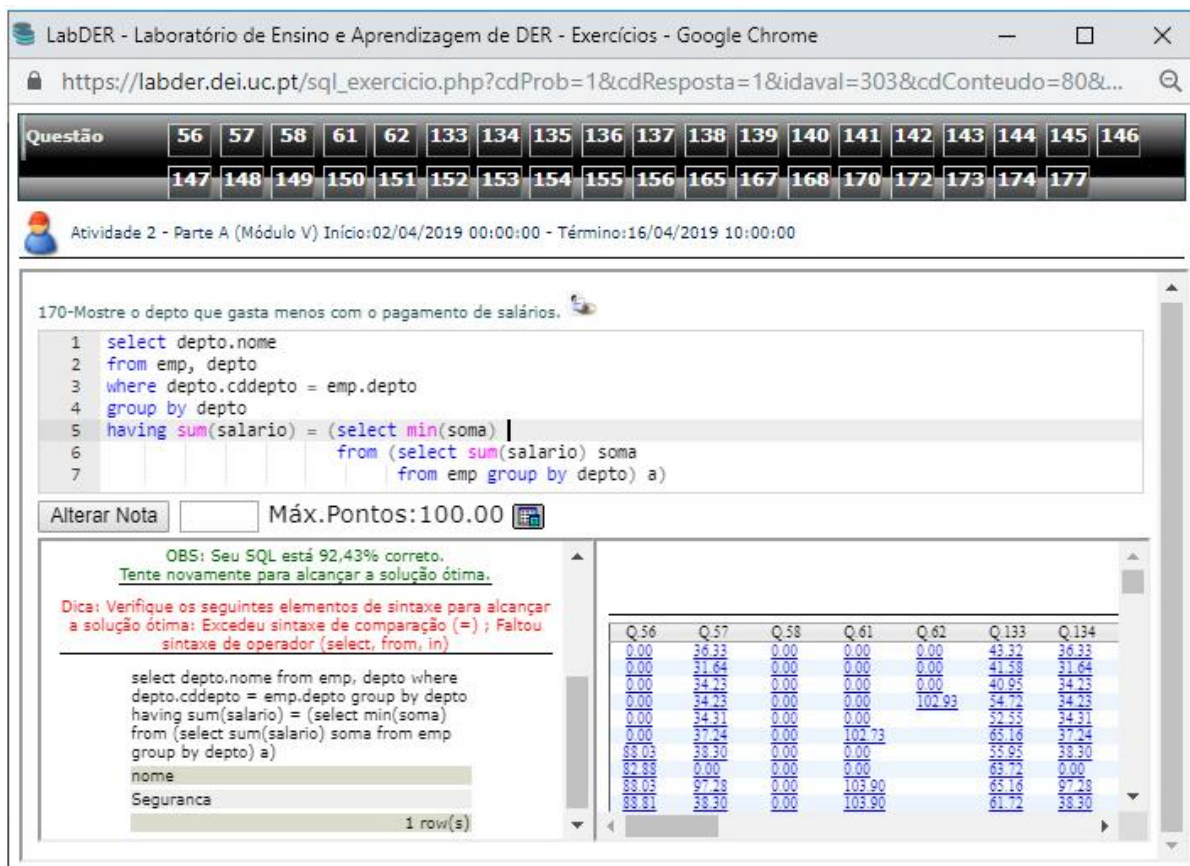


Figura 6-45. Interface do professor que avalia o SQL da resposta do aluno.

O professor pode alterar ou concordar com a avaliação automática atribuída pelo *framework* (ver *feedback* em verde 92,43% na Figura 6-45).

6.4.3.4 Resultados - avaliação de escolha múltipla

O resultado a ser analisado é a resposta do aluno à questão 76. A Figura 6-46 mostra a interface do professor ao avaliar a resposta de escolha múltipla de um aluno. A solução do

aluno para a questão 76, quando comparada com a resposta do sistema, foi classificada como correta. Portanto, a área de *feedback* notifica o utilizador que a opção de escolha múltipla está correta com um *feedback* “Solução Correta - Continue Assim”.

The screenshot shows a web browser window titled "LabDER - Laboratório de Ensino e Aprendizagem de DER - Exercícios - Mozilla Firefox". The address bar shows a URL with parameters for question ID, response ID, and user ID. The interface displays a question about referential integrity constraints. The question text is: "76-Numa base de dados relacional, examine as afirmações sobre a restrição de integridade referencial: (I) Ao inserir um registro numa tabela, os valores das chaves estrangeiras devem existir como valores na chave primária referenciada ou serem nulos; (II) Ao alterar registros numa tabela, os novos valores para chaves estrangeiras devem existir como valores nas chaves primárias referenciadas ou serem nulos; (III) Ao excluir registros numa tabela os valores das chaves primárias não podem existir em outros registros como valores em chaves estrangeiras; Assinale a opção correta: (Máximo de Tentativas: 1)". The options are: (a) somente a (I) é verdadeira; (b) somente a (II) é verdadeira; (c) somente a (III) é verdadeira; (d) somente a (II) e (III) são verdadeiras; (e) todas (I), (II) e (III) são verdadeiras. Option (e) is selected and marked as correct with a green checkmark. Below the question, there is a button "Alterar Nota" and a field "Máx.Pontos: 1.00". To the right, a "Histórico de Respostas" table shows scores for various questions. The table has columns for question IDs (Q.45, Q.48, Q.52, Q.54, Q.55, Q.69, Q.76, Q.100, Q.102, Q.111) and rows for individual scores and average scores (MP).

Histórico de Respostas									
Q.45	Q.48	Q.52	Q.54	Q.55	Q.69	Q.76	Q.100	Q.102	Q.111
100.00	0.00	100.00	102.21	103.08	1.00	1.00	87.54	101.36	1.00
	0.00						89.25		
	98.36						0.00		
	97.99						89.25		
	98.36						94.79		
MP	MP	MP	MP	MP	MP	MP	MP	MP	MP
100.00	98.36	100.00	102.21	103.08	1.00	1.00	94.79	101.36	1.00

Figura 6-46. Interface professor que avalia a resposta de escolha múltipla do aluno.

O professor pode alterar a avaliação automática, e a nova nota é guardada ficando disponível no painel do histórico de resposta do aluno. Raramente o professor altera uma nota de uma questão de escolha múltipla, mas surgem situações em que a questão está mal formulada ou há o registo de uma solução incorreta.

6.4.3.5 Resultados - avaliação de questão de texto livre

O resultado a ser analisado é a resposta do aluno à questão 66. A Figura 6-47 mostra a interface do professor ao avaliar a resposta de texto livre de um aluno. A solução do aluno para a questão 66, quando comparada com a resposta do sistema, foi automaticamente avaliada numa escala de 0 a 100, através de um algoritmo de similaridade. Os alunos e professores são sempre avisados que o algoritmo atribui um valor provisório, mas que é o professor que atribui a classificação final.

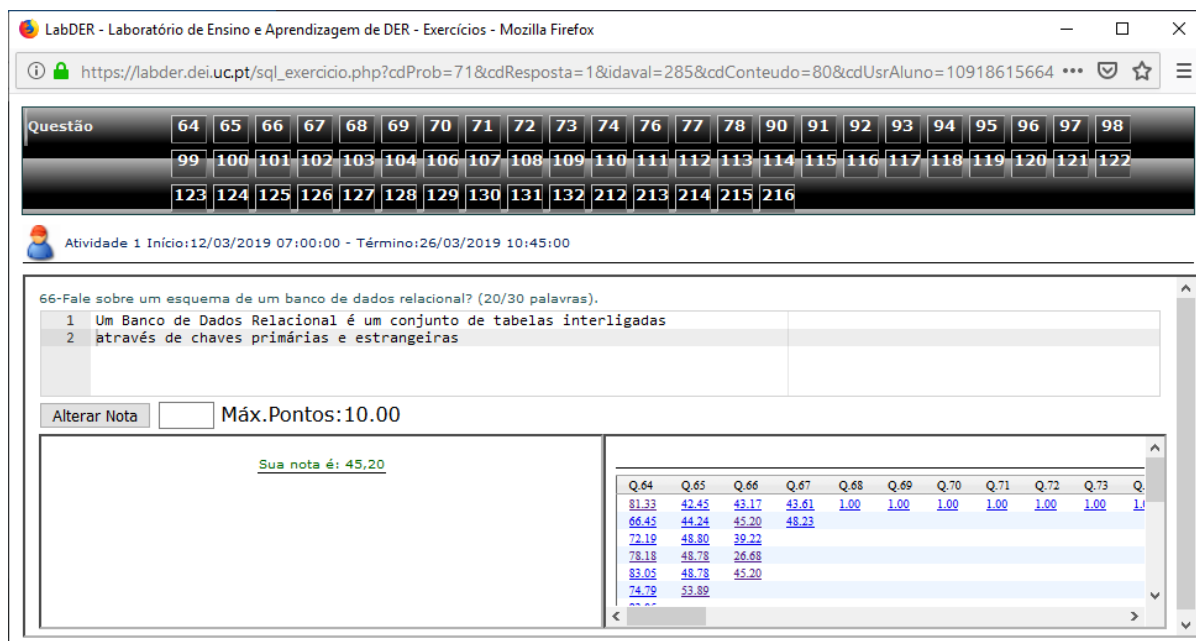


Figura 6-47. Interface do professor que avalia a resposta de texto livre do aluno.

O professor pode alterar a avaliação automática, e a nova nota é guardada ficando disponível no painel do histórico de resposta do aluno. Normalmente, o professor altera as notas das questões de texto livre, pois o algoritmo de similaridade ainda não é muito preciso.

Os alunos e professores são sempre avisados sobre a limitação do algoritmo de similaridade, entretanto, os alunos gostam e tentam, por diversas vezes, melhorar a nota obtida. Por exemplo, as questões 64, 65 e 66 são de texto livre e o painel de histórico mostra as submissões feitas pelo aluno. Também existem mais respostas que podem ser vistas na barra de rolagem vertical no painel de histórico (ver Figura 6-47).

6.4.3.6 Resultados - avaliação do professor

A avaliação dos professores foi recolhida através de um inquérito online desenvolvido no Google Forms, após a entrega das notas finais dos alunos e o encerramento das atividades acadêmicas, junto da coordenação do curso do ISCAC.

O questionário é do tipo semiaberto e possui três dimensões: a primeira dimensão visa avaliar o LabDER como um ambiente de trabalho virtual de ensino e aprendizagem de bases de dados relacionais; a segunda dimensão procura revelar se o professor aceita ou rejeita a avaliação automática do DER; a terceira dimensão visa identificar a utilidade da avaliação automática em sala de aula. As questões de 1 a 7 são fechadas com as opções do tipo “sim” e “não” e, adicionalmente, com a opção de resposta aberta “outro”, sendo a questão 8 do tipo aberta (ver Tabela 6-14).

Tabela 6-14. Questionário de avaliação do LabDER aplicado aos professores.

#	Dimensão	Conteúdo
1	Ava	O LabDER contempla os principais conceitos de BD relacionais?
2	Ava	O LabDER auxiliou a cumprir as atividades do plano de ensino da disciplina?
3	Ava	Recomendaria o LabDER para o próximo semestre?
4	Avaliação Automática	Concorda com a nota sugerida pelo LabDER acerca das respostas dos alunos do tipo diagrama ER?
5	Avaliação Automática	Na sua opinião, o <i>feedback</i> semântico do LabDER ajudou os alunos a melhorar o desenvolvimento das respostas de DER?
6	Utilidade	O LabDER diminuiu o tempo de avaliação de exames e exercícios de diagrama ER?
7	Utilidade	Ao considerar o acompanhamento das respostas dos alunos, o LabDER facilitou o seu trabalho no diagnóstico das fraquezas da turma?
8	Utilidade	Por favor, adicione quaisquer informações que considere relevantes para a melhoria do LabDER como ambiente de ensino de BD.

Resultados

A dimensão “Ava” descrita na Tabela 6-14 contém três questões que foram unânimes em classificar positivamente o LabDER. Na opinião dos professores, o LabDER cumpre com seu papel como ambiente virtual de ensino e aprendizagem de bases de dados relacionais. Nas reuniões com os professores, a percepção era a de que o plano de ensino executado se assemelhou ao dos cursos de bacharelato em Ciências Informáticas e Engenharia Informática, devido à profundidade oferecida pelos tópicos e que, nas próximas edições, seria reduzida a quantidade de atividades.

A dimensão “Avaliação Automática” descrita na Tabela 6-14 verifica se os professores concordam com a nota sugerida pelo ambiente LabDER acerca das respostas do tipo DER submetidas pelos alunos. Os professores responderam positivamente pois concordam com a nota fornecida pelo ambiente. Ambos os professores concordaram que o *feedback* semântico do LabDER ajudou os alunos a aperfeiçoar o desenvolvimento das respostas de DER.

A dimensão “Utilidade” descrita na Tabela 6-14 contém três questões, os professores concordam que o sistema tanto diminui o tempo de correção de exames e exercícios, como ajuda no diagnóstico das fraquezas da turma. Os seguintes comentários foram adicionados pelos professores na questão aberta:

“A inserção de novas questões poderia ser mais simples.”

“A utilização do LabDER autonomiza a aprendizagem do aluno de forma relevante, porém, exige que o docente domine bem a plataforma, para que tudo decorra na normalidade, e que reflita bem sobre a forma como irá realizar a avaliação.”

Estas respostas podem estar associadas ao excesso de carga de trabalho que tiveram ao registrar questões e suas respectivas respostas, no ambiente. Esse trabalho manual de registo de

questões e respostas poderá provavelmente ser aperfeiçoado com um mecanismo importado de ficheiros do Microsoft Word, por exemplo.

Por um lado, o sistema é muito flexível, e isso é bom para o professor que pode mudar sempre o formato da avaliação sem mudar o plano de ensino. Por outro lado, é necessário avaliar o que está a ser exigido aos alunos. A atividade de criar uma avaliação formal no LabDER tem o mesmo peso de criar uma avaliação em papel, ou seja, deve-se evitar o erro, para não ter que correr o risco de anular questões ou criar uma nova avaliação.

O comentário do professor sobre a utilidade do ambiente merece destaque “*A utilização do LabDER autonomiza a aprendizagem do aluno de forma relevante*”. Entendemos que o LabDER atinge o seu objetivo de avaliação automática de DER, uma vez que os professores concordam positivamente com as dimensões sobre o uso do LabDER como um AVA, e com as utilidades da avaliação automática do DER, em sala de aula. Bem como, definem que o LabDER fornece autonomia da aprendizagem. Essa autonomia é justamente o resultado do modelo avaliação automática.

O LabDER foi utilizado de forma intensiva pelos alunos do ISCAC: foram mais de 15 mil respostas submetidas e automaticamente avaliadas pelo ambiente de trabalho, e outras 15 mil em modo de treino. Foram apresentadas as evidências quantitativas para aferir a participação do aluno no ambiente de trabalho, e assim responder ao primeiro objetivo da avaliação. Por analogia, verificou-se que os alunos do ISCAC tiveram uma participação 4 vezes maior do que é normalmente executado, noutra ambiente de trabalho da mesma categoria. Foram ainda apresentadas as evidências da participação dos alunos na utilização das tecnologias atuais do Google Analytics que foram monitorizadas via internet.

Adicionalmente, para medir o desempenho dos alunos na disciplina, foi criado um relatório sucinto dos desempenhos dos alunos. O resultado foi um mapeamento completo de todos os conceitos ensinados de DER, com a frequência e percentagem de acerto de cada componente.

Por fim, o resultado do inquérito feito aos professores mostra que a avaliação automática de DER através do LabDER pode ser utilizada satisfatoriamente em sala de aula. Os professores responderam de forma positiva a todo o questionário. Pelas respostas dos professores pode interpretar-se que a avaliação automática do LabDER é eficiente no que se propõe a fazer.

Futuras evidências podem ser planejadas e recolhidas para aferir com mais exatidão o quão eficaz a abordagem de avaliação automática pode ser demonstrada através da sua utilização em sala de aula. As evidências da utilização do LabDER no ISCAC demonstram um desempenho dos alunos em cada conceito de BD (ver Tabela 6-13) cerca de 20% superior, quando comparado com outras 6 instituições de ensino participantes no projeto, e cerca de 60% a 70% superior, no conhecimento específico de cardinalidade e participação. O projeto de investigação não visava a comparação de desempenho de alunos e, por isso não foi abordado nesta tese.

7.1 Considerações iniciais

O ambiente virtual LabDER com um *framework* inovador de avaliação automática foi desenvolvido e apresentado e, através de testes de funcionalidade e desempenho e usabilidade, o sistema foi implementado corretamente, podendo ser usado até 1.000 utilizadores em simultâneo, acedendo de diferentes partes do mundo, com um tempo de resposta satisfatório. Os objetivos propostos para automatizar a avaliação de DER, no contexto de ensino e aprendizagem de Bases de Dados Relacionais, foram cumpridos. Para esse efeito, seguiu-se a metodologia de investigação DSRM, como guia, no desenvolvimento, demonstração e validação dos artefactos.

A primeira fase consistiu na identificação dos problemas e limitações das abordagens (soluções) encontradas na literatura, para delinear os objetivos e propor uma nova abordagem, conforme o capítulo 2.

A segunda fase dessa investigação, foi projetar um *framework* inovador que utilizasse métricas de engenharia de software e teoria de compiladores para colocar, ao dispor do utilizador, o melhor *feedback* na avaliação de DER, descrito no capítulo 3.

A terceira fase visou desenvolver e testar um ambiente virtual para o ensino de bases de dados relacionais, denominado LabDER, que integrasse o *framework* de avaliação automática para recolher os DER e as avaliações dos professores, conforme capítulo 4.

Por fim, os dados recolhidos foram treinados e testados até obter o modelo de avaliação automática de DER que mais se aproximasse da avaliação do professor. O LabDER foi validado pelos utilizadores e atendeu aos requisitos propostos, conforme descrito no capítulo 5.

7.2 Contribuições e reflexões

A solução desenvolvida destaca-se inicialmente pela investigação sobre como medir o diagrama de entidade e relacionamento, e apresenta um conjunto de métricas específicas para avaliar os conceitos de DER, usualmente ensinados na disciplina de bases de dados. As métricas quantitativas oferecem métodos para comparar estruturas dos DER e fornecem indicadores fiáveis ao medir a distância entre a resposta do aluno e a resposta do professor.

A principal limitação, encontrada nas abordagens anteriores, está na análise da resposta do aluno, classificada aqui como resposta restritiva. Esta limitação não permite ao aluno responder mais de uma vez, ou aperfeiçoar a solução, como ocorre num processo natural de desenvolvimento de soluções computacionais com tentativa e erro. Para superar essa limitação, implementámos a análise da resposta do aluno inspirada nas fases de análise de compiladores. A análise compreende as fases de análises léxicas, sintáticas e semânticas que auxiliam os utilizadores na tarefa de identificar e corrigir o erro. A análise baseada em compiladores oferece as seguintes vantagens: (a) análise flexível que avalia diferentes estruturas, como diagramas ER e código SQL; (b) identifica a existência de erro em três níveis: léxico, sintático e semântico; (c) implementa diferentes estratégias de *feedback*, por exemplo, mensagens genéricas ou específicas, e ainda associadas a conceitos de BD ou formato de dicas; (d) implementa equivalência entre estruturas computacionais; (e) auxilia na tarefa de verificar se a resposta do aluno é mais eficiente do que a resposta do professor, pelas métricas de engenharia de software; (f) dá prioridade ao tratamento de erros, identificando, rapidamente, os erros existentes na solução do aluno, a partir da resposta do professor; (g) dá um *feedback* semântico, que identifica o erro e permite o desenvolvimento de novas soluções computacionais, seja para o caso de corrigir um erro, ou aperfeiçoar a solução.

O *feedback* semântico inclui informações resultantes da análise do compilador, métricas de engenharia de software, e uma nota final baseada no modelo preditivo, que tem como entrada as métricas extraídas das respostas do aluno e do professor. O melhor modelo preditivo de avaliação de DER foi encontrado no terceiro ciclo do desenvolvimento do CRISP-DM. Em cada ciclo, foram executados nove ensaios com dez classificadores, ajustados com uma validação cruzada de dez vezes. Assim, foram construídos 900 modelos por ciclo, e avaliados quanto às medidas de desempenho para classificadores binários. O algoritmo RFC foi certamente o mais adequado para prever as avaliações no LabDER, pois obteve o melhor desempenho médio dentre os demais classificadores. A reestruturação dos atributos foi determinante para atingir o critério mínimo de 90% de *accuracy*, pois implicou um aumento

de até 30% do desempenho de alguns classificadores. Os algoritmos, de aprendizagem computacional supervisionado, provaram ser úteis para disponibilizar um modelo preditivo e depois usá-lo para prever a nota do aluno.

No LabDER, esta abordagem de avaliação automática permitiu: (a) aceitar múltiplas respostas corretas; (b) prever as notas finais de cada resposta do aluno que são geradas a partir do modelo preditivo que mede a distância entre as respostas do aluno e do professor, através das métricas associadas à conceitos de BD; (c) auxiliar os alunos e professores a identificar, mais rapidamente, o erro da solução submetida através do *feedback* que associa erros a conceitos de BD; (d) oferecer ao aluno a possibilidade de refazer a solução; (e) oferecer ao professor um sistema que pode ser usado em avaliações diagnósticas, formativas e sumativas.

A avaliação automática traz muitas vantagens para o contexto de sala de aula. Na perspectiva do professor, fornece suporte computacional para avaliar e acompanhar os alunos, isto auxilia no trabalho e despende menos tempo em atividades repetitivas. E, obviamente, o facto de ter o apoio computacional para corrigir automaticamente todas as respostas dos alunos, é um imenso alívio. Adicionalmente, o *feedback* auxilia o professor a perceber quais os conceitos em que os alunos estão a ter mais dificuldade em aprender e auxilia-o a planear as aulas, para melhorar o processo de aprendizagem. O professor percebe os obstáculos que os alunos, eventualmente, enfrentam mas de modo diferente de outros cenários pois o aluno não fica com vergonha de errar ou de expor a sua dificuldade. Isso ocorre porque no LabDER, errar faz parte do processo de aprendizagem. Na perspectiva dos alunos, não há problema em errar num sistema que só aceita a melhor resposta, tentar superar a resposta do professor é um desafio que o motiva a continuar na procura da melhor solução. O *framework* foi construído para esse efeito, que inclui uma aprendizagem mais autónoma para o aluno e uma forma de lecionar mais personalizada para o professor.

Os relatórios do LabDER podem ser utilizados pelo professor como um sistema de apoio à decisão e de recomendação. A experiência mais reveladora do potencial do *framework* foi testada na turma do ISCAC, durante a reunião com o professor colaborador. Foi exposto ao professor o relatório de erros mais frequentes dos alunos, e comprovada uma demasiada frequência de erros cometidos pelos alunos sob o conceito de cardinalidade. A decisão tomada foi a de dedicar uma aula de reforço sobre o tema de cardinalidade, utilizando exercícios na plataforma e vários exemplos didáticos. Após a aula de reforço, os alunos continuaram o desenvolvimento dos exercícios, tanto online como em sala de aula. Na reunião seguinte, através dos relatórios do LabDER, foi diagnosticada uma redução significativa de erros

cometidos pelos alunos, no conceito de cardinalidade. Normalmente o professor repete um tópico de aula a pedido dos alunos, porém como o LabDER forneceu informações ao professor para tomada de decisão, este foi proativo ao revisitar o tópico de aula. Os resultados mostraram o efeito positivo da aula de intervenção, inicialmente diagnosticada através do *feedback* emitido nos relatórios da ferramenta.

No final do projeto, 98% dos utilizadores avaliaram a usabilidade do sistema como aceitável, a maioria avaliou como “Excelente” na escala do SUS. O sucesso do uso da plataforma dá-se por várias razões, sendo a primeira o planeamento do curso. A ideia do plano de atividades era evitar tirar o professor da zona de conforto com o material de que tem conhecimento. Por isso, o plano de atividades era feito sob medida, usando como requisito o plano de ensino do professor. O segundo fator, deve-se ao interesse dos professores que realmente participaram no projeto com muito entusiasmo e dedicação. Os professores estavam 100% sincronizados com o investigador que atuava, sempre que necessário, para qualquer dúvida ou mesmo apoio nas aulas quando solicitado para. As reuniões presenciais não foram apenas para esclarecer dúvidas, mas principalmente para expor como tirar proveito da avaliação automática, de forma a manter os alunos motivados. E aqui está o fator principal, isto é, os alunos mantiveram-se motivados pois aprenderam a usar o *feedback* da avaliação automática para tirar mais dúvidas com os professores.

As demais limitações das abordagens anteriores foram todas superadas. Uma API para desenho de diagramas ER foi incorporada no LabDER, e permite que os alunos e professores desenhem livremente, utilizando funções de clicar e arrastar do rato, solucionando o problema de interface restritiva. O ambiente aceita múltiplas respostas corretas e não está limitado ao *feedback* do tipo certo ou errado, evitando assim que os alunos submetam uma resposta restritiva. O LabDER não limita o aluno a refletir sobre o problema a ser resolvido, como nas abordagens anteriores com o problema dos destaques visuais que fornecem a resposta esperada do sistema na descrição da questão. A solução adotada é baseada nas questões tipicamente usadas em livros de BD, e oferece ao professor três cenários para criar questões no sistema. A notação *crow's foot* elimina o problema da notação restritiva ou desconhecida.

7.3 Limitações da investigação

Uma limitação está associada à peculiaridade da avaliação. Não foi encontrado um modelo de predição, no primeiro ciclo do CRISP-DM, como consequência dos diferentes critérios de avaliação que os professores usam para atribuir uma nota. Foi necessária uma

intervenção para propor uma forma de contabilizar os conceitos de BD que reduzisse a discrepância das avaliações atribuídas pelos professores. O modelo atual de predição é resultante do consenso dos professores para o ensaio, entretanto não avaliámos se este modelo pode, de facto, ser aceite por outros professores.

Existem ainda outras limitações técnicas encontradas no LabDER. A primeira limitação está relacionada com as diferentes sintaxes de SQL usadas pelos SGBD. Apesar dos esforços da ANSI em padronizar a linguagem SQL, ainda existem distinções nas sintaxes de comandos DQL, DML, DDL e DCL. A API de desenho de DER exporta o diagrama para diferentes SGBD, entretanto o LabDER está sob a plataforma MySQL, e por isso não suporta particularidades de sintaxe de outros SGBD. Muitos professores despendem anos a aperfeiçoar o material didático para que possa ser usado numa plataforma específica, como Microsoft SQL Server ou Postgree. Alguns professores não aceitaram participar do projeto, justamente por causa dessa limitação de sintaxe. Isso ocorre porque diversos tópicos de BD, como procedimentos, gatilhos, visão, transações são difíceis de ensinar e é onde existem muitas diferenciações de sintaxe entre os SGBD. Os professores que rejeitaram o projeto tinham demonstrado muito interesse pela avaliação automática e por todos os benefícios do ensino virtual. No entanto, isso não foi o suficiente para mudar o material didático que tinham, no mínimo com 15 anos de maturidade, noutra plataforma.

Adicionalmente, o LabDER foi concebido para a construção de DER na notação *crow's foot*, portanto, mesmo que o *framework* seja genérico e ofereça uma estrutura para aplicação noutros diagramas, ainda é dispendioso automatizar outras notações. É necessário encontrar ou implementar APIs que ofereçam a edição de diagramas online, bem como identificar os conceitos relacionados com a notação e as respetivas métricas e que as afirmem. Por fim, existe a necessidade de encontrar um modelo preditivo que se adeque à avaliação de todos os professores.

7.4 Trabalhos futuros

Há questões ainda não resolvidas que servem de motivações para aprimorar o desenvolvimento do LabDER e avançar na área de avaliação automática.

Sendo comum o uso de outras notações diagramáticas no contexto de ensino de bases de dados, seria interessante investigar a capacidade do *framework* para avaliar outras notações de BD como IDEF1X. Para isso, além de encontrar uma API que forneça suporte para edição

destas notações, será necessário investigar novas métricas que auxiliem na medição dos conceitos de cada uma das propriedades da notação selecionada. Assim como é, igualmente, necessário encontrar o melhor algoritmo para prever as notas finais dos alunos.

Durante as entrevistas com os professores notámos que a maioria recorria a um projeto de BD para avaliar os alunos, que era aplicado no fim do curso, e geralmente em grupo. É preciso evoluir o sistema para permitir outras modalidades de avaliação, como no caso do projeto de BD. Anteriormente, investigámos como avaliar automaticamente questões de texto livre, o que se revelou um grande desafio. Alguns algoritmos poderiam ser aplicados para descobrir plágio entre os trabalhos académicos e fornecer alguns indicadores para que o professor possa avaliar mais rapidamente. Outra linha de investigação, baseada na análise de requisito e processamento de linguagem natural, poderia oferecer um modelo prévio ou sistema, e assim encontrar formas de comparar e medir o projeto submetido pelo aluno.

O LabDER ainda não oferece a opção de personalizar os critérios de avaliação, por exemplo, ignorar a avaliação do conceito de cardinalidade. Uma alteração no sistema poderia oferecer uma funcionalidade para o professor definir quais os conceitos de BD que gostaria de avaliar por teste. Entretanto, isso não seria o suficiente, há que ter em conta que o modelo preditivo não seria capaz de prever adequadamente. Seria necessário investigar como superar essa limitação, pois não parece uma solução adequada desenvolver mais de um modelo preditivo por professor ou por critérios de avaliação.

Podemos aperfeiçoar o LabDER para partilhar as questões entre os professores. As questões padrão do LabDER são partilhadas e sincronizadas manualmente pelo investigador. Entretanto, não existem procedimentos específicos de partilha e sincronização de respostas entre os professores. As questões são úteis para o professor porque já foram validadas inúmeras vezes por alunos e outros professores, dentro do ambiente académico.

Também podemos propor investigações relacionadas com sistemas de suporte à decisão, para recomendar ao professor as melhores estratégias de ensino e aos alunos recomendações específicas de estudo para atingir a nota máxima. São armazenadas na BD do LabDER todas as interações dos utilizadores, e esses dados podem ser analisados com algoritmos inteligentes para propor recomendações. Seria interessante investigar se os alunos e professores poderiam evitar erros ou aprender de forma mais eficiente, recorrendo ao histórico de respostas.

Tópicos como álgebra relacional, performance de BD, índices, transações ainda são um desafio para automatizar o ensino de bases de dados relacionais. Existem alguns componentes de edição de álgebra relacional, que fornecem um código SQL, poderíamos investigar como

incorporar e avaliar álgebra relacional. Uma vez que se avança sobre a avaliação da álgebra, poderíamos seguir para o tópico de desempenho e índices de BD. Alguns livros de BD usam álgebra relacional para explicar como aumentar o desempenho de consultas, reduzindo o tamanho da árvore, e mostrando os custos da complexidade de uma consulta. Seria interessante investigar como avaliar essas estruturas com o foco na complexidade do desempenho de BD. Uma série de métricas existentes podem ser revalidadas ou descobertas outras mais eficientes.

O LabDER terá mais aceitabilidade pelos professores quando conseguir ser utilizado em diversos SGBD. Para isso, o sistema precisa de ser adaptado a outras plataformas e um esforço precisa ser feito, no desenvolvimento de sistemas mais adaptativos.

Por fim, acreditamos que todos os conceitos de ensino de bases de dados são capazes de serem virtualizados com novos recursos visuais, e com o apoio da avaliação automática baseado em mecanismos de IA explicável. Vislumbram-se novas formas de interação homem computador capazes de explicar de forma transparente os resultados da tomada de decisão da máquina.

Publicações resultantes deste trabalho de investigação

- Lino, A., Rocha, Á., Macedo, L., & Sizo, A. (2019). LabDER-Relational Database Virtual Learning Environment. In *38th International Conference on Conceptual Modeling (ER 2019)* (Vol. 2469, pp. 42–55). Salvador, Brazil: CEUR-WS.org.
- Lino, A., Rocha, Á., Macedo, L., & Sizo, A. (2019). Application of clustering-based decision tree approach in SQL query error database. *Future Generation Computer Systems*, 93. <https://doi.org/10.1016/j.future.2018.10.038>
- Lino, A., Rocha, Á., & Macedo, L. (2018). A virtual learning environment to evaluate entity relationship diagram automatically. In *Pattern Recognition (RecPad)*. Coimbra, Portugal: Associação Portuguesa de Reconhecimento de Padrões.
- Del Pino Lino, A., & Rocha, A. (2018). Automatic evaluation of ERD in e-learning environments. In *Iberian Conference on Information Systems and Technologies, CISTI* (Vol. 2018-June). <https://doi.org/10.23919/CISTI.2018.8399401>
- Lino, A., Rocha, Á., & Sizo, A. (2017). Virtual teaching and learning environments: automatic evaluation with artificial neural networks. *Cluster Computing*, 1–11. <https://doi.org/10.1007/s10586-017-1122-y>
- Lino, A., Rocha, Á., & Sizo, A. (2016). Virtual teaching and learning environments: Automatic evaluation with symbolic regression. *Journal of Intelligent & Fuzzy Systems*, 31(4), 2061–2072. <https://doi.org/10.3233/JIFS-169045>
- Lino, A., Rocha, Á., & Sizo, A. (2016). A Proposal for Automatic Evaluation by Symbolic Regression in Virtual Learning Environments. In *New Advances in Information Systems and Technologies. Advances in Intelligent Systems and Computing* (444th ed., pp. 855–865). Springer International Publishing. https://doi.org/10.1007/978-3-319-31232-3_81

Referências Bibliográficas

- ACM, & IEEE. (2013). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. (ACM Computing Curricula Task Force, Ed.). New York, NY, USA: ACM. <https://doi.org/10.1145/2534860>
- Ahadi, A., Behbood, V., Vihavainen, A., Prior, J., & Lister, R. (2016). Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education - SIGCSE '16* (pp. 401–406). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2839509.2844640>
- Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. In *Proceedings - 39th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2013*. <https://doi.org/10.1109/SEAA.2013.28>
- Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2007). *Compilers: Principles, Techniques, and Tools (2nd Edition)* (2nd Editio). Boston, MA, USA: Pearson/Addison Wesley.
- Ammann, P., & Offutt, J. (2017). *Introduction to software testing*. (Cambridge University Press, Ed.) (2nd Edition).
- Antony, S. R., & Batra, D. (2002). CODASYS: a consulting tool for novice database designers. *ACM SIGMIS Database: The DATABASE for Advances in Information Systems*, 33(3), 54–68. <https://doi.org/10.1145/569905.569911>
- Ashraf, J., Iqbal, N., Khattak, N. S., & Zaidi, A. M. (2010). Speaker independent Urdu speech recognition using HMM. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6177 LNCS, pp. 140–148). https://doi.org/10.1007/978-3-642-13881-2_14
- Bachman, C. W. (1969). Data structure diagrams. *ACM SIGMIS Database*, 1(2), 4–10. <https://doi.org/10.1145/1017466.1017467>
- Bai, L., Meredith, R., & Burstein, F. (2018). A data quality framework, method and tools for managing data quality in a health care setting: an action case study. *Journal of Decision Systems*, 27(sup1), 144–154. <https://doi.org/10.1080/12460125.2018.1460161>
- Baker, E. L., Barton, P. E., Darling-Hammond, L., Haertel, E., Ladd, H. F., Linn, R. L., ... Shepard, L. A. (2000). *Problems with the Use of Student Test Scores to Evaluate Teachers*. Washington, DC. Retrieved from www.epi.org
- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3), 114–123.
- Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human-Computer Interaction*, 24(6), 574–594. <https://doi.org/10.1080/10447310802205776>
- Barker, K. (1992). A logical model and schema integration architecture for determining

- equivalence in ER-diagrams. In *Proceedings ICCI '92: Fourth International Conference on Computing and Information* (pp. 320–324). Toronto, Canada: IEEE Comput. Soc. Press. <https://doi.org/10.1109/ICCI.1992.227645>
- Basili, V. R., & Rombach, H. D. (1988). The TAME project: Towards improvement-oriented software environments. *Software Engineering, IEEE Transactions On*, 14(6), 758–773.
- Batini, C., & Scannapieco, M. (2016). *Data and Information Quality*. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-24106-7>
- Batra, D., & Antony, S. R. (1994). Novice errors in conceptual database design. *European Journal of Information Systems*, 3(1), 57–69. <https://doi.org/10.1057/ejis.1994.7>
- Batra, D., Hoffler, J. A., & Bostrom, R. P. (1990). Comparing representations with relational and EER models. *Communications of the ACM*, 33(2), 126–139. <https://doi.org/10.1145/75577.75579>
- Batra, Dinesh. (2007). Cognitive complexity in data modeling: Causes and recommendations. *Requirements Engineering*, 12(4), 231–244. <https://doi.org/10.1007/s00766-006-0040-y>
- Batra, Dinesh, & Davis, J. G. (1992). Conceptual data modelling in database design: similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies*, 37(1), 83–101. [https://doi.org/10.1016/0020-7373\(92\)90092-Y](https://doi.org/10.1016/0020-7373(92)90092-Y)
- Beck, K. (2003). *Test Driven Development: By Example* (1st Edition). Addison-Wesley.
- Berdanier, C. G. P., Baker, E., Wang, W., & McComb, C. (2019). Opportunities for natural language processing in qualitative engineering education research: Two examples. In *Proceedings - Frontiers in Education Conference, FIE* (Vol. 2018-October). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/FIE.2018.8658747>
- Bevan, N., Barnum, C., Cockton, G., Nielsen, J., Spool, J., & Wixon, D. (2003). The “magic number 5”: is it enough for web testing? In *CHI '03 extended abstracts on Human factors in computing systems - CHI '03* (p. 698). New York, New York, USA: ACM Press. <https://doi.org/10.1145/765891.765936>
- Bligh, B. (2007). *Formative computer based assessment in diagram based domains*. University of Nottingham. Retrieved from <http://eprints.nottingham.ac.uk/13569/>
- Blom, M. (2010). Is scrum and XP suitable for CSE development? In *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2010.04.168>
- Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2), 245–271. [https://doi.org/10.1016/s0004-3702\(97\)00063-5](https://doi.org/10.1016/s0004-3702(97)00063-5)
- Bluman, A. G. (2017). *Elementary Statistics: A Step By Step Approach* (10th ed.). McGraw-Hill Education.
- Boehm, B. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), 22–42. <https://doi.org/10.1145/12944.12948>
- Bogdanović, M., Stanimirović, A., Davidović, N., & Stoimenov, L. (2008). The Development and Usage of a Relational Database Design Tool for Educational Purposes. In *Informing Science & IT Education Conference (InSITE'08)* (pp. 251–258).
- Boney, B. (2010). *Feature selection based on information theory*. University of Alicante.
- Bonissone, P. P. (1980). A fuzzy sets based linguistic approach: Theory and applications. In *WSC '80 Proceedings of the 12th conference on Winter simulation* (pp. 99–111). Orlando, FL: IEEE Press.
- Boris, J., Nenad, J., & Svetlozar, N. (2013). Process and Data Logic Integration: Logical Links

- between UML Use Case Narratives and ER Diagrams. *Journal of Computing and Information Technology*, 21(3), 161–170. <https://doi.org/https://doi.org/10.2498/cit.1002192>
- Braunmuller, B., Ester, M., Kriegel, H.-P., & Sander, J. (2001). Multiple similarity queries: a basic DBMS operation for mining in metric databases. *IEEE Transactions on Knowledge and Data Engineering*, 13(1), 79–95. <https://doi.org/10.1109/69.908982>
- Brooke, J. (1996). SUS: A “Quick and Dirty” Usability Scale. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. Weerdmeester (Eds.), *Usability Evaluation In Industry* (pp. 189–194). London: Taylor & Francis. <https://doi.org/https://doi.org/10.1201/9781498710411>
- Bruce, T. A., & A., T. (1992). *Designing quality databases with IDEF1X information models*. New York, NY, USA: Dorset House Pub.
- Burnham, K. P., & Anderson, D. R. (1998). *Model Selection and Inference: A Practical Information-Theoretic Approach*. *Model Selection and Inference*. Springer New York. <https://doi.org/10.1007/978-1-4757-2917-7>
- Burnstein, I. (2003). *Practical Software Testing: A Process-Oriented Approach* (Springer P). New York, USA: Springer New York.
- Bushra, J., Javed, F., Asma, B., & Shaista, G. (2010). Empirical validation of relational database metrics for effort estimation. In *INC 2010: 6th International Conference on Networked Computing* (pp. 1–5). Gyeongju, South Korea: IEEE. Retrieved from <https://ieeexplore.ieee.org/document/5484854>
- Cagiltay, N. E., Tokdemir, G., Kilic, O., & Topalli, D. (2013). Performing and analyzing non-formal inspections of entity relationship diagram (ERD). *Journal of Systems and Software*, 86(8), 2184–2195. <https://doi.org/10.1016/J.JSS.2013.03.106>
- Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300, 70–79. <https://doi.org/10.1016/j.neucom.2017.11.077>
- Calero, C., & Piattini, M. (2002). Metrics for databases: a way to assure the quality (pp. 57–83). Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0831-1_4
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Rüdiger, W. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. CRISP-DM consortium. Retrieved from <https://www.the-modeling-agency.com/crisp-dm.pdf>
- Chen, P. P.-S., & Pin-Shan, P. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36. <https://doi.org/10.1145/320434.320440>
- Chen, P. P. S. (1983). English sentence structure and entity-relationship diagrams. *Information Sciences*, 29(2–3), 127–149. [https://doi.org/10.1016/0020-0255\(83\)90014-2](https://doi.org/10.1016/0020-0255(83)90014-2)
- Chen, X., Mitrovic, A., & Mathews, M. (2018). Exploring Adaptive Strategies for Providing Learning Activities. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization - UMAP '18* (pp. 139–145). New York, NY, USA: ACM Press. <https://doi.org/10.1145/3209219.3209221>
- Chen, Y., Miao, D., & Wang, R. (2010). A rough set approach to feature selection based on ant colony optimization. *Pattern Recognition Letters*, 31(3), 226–233. <https://doi.org/10.1016/j.patrec.2009.10.013>
- Cheng, P. C.-H., Lowe, R. K., & Scaife, M. (2001). Cognitive Science Approaches To Understanding Diagrammatic Representations. In *Thinking with Diagrams* (pp. 79–94). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-017-3524-7_5

- Choppella, V., Sengupta, A., Robertson, E. L., & Johnson, S. D. (2007). Preliminary explorations in specifying and validating entity-relationship models in PVS. In *Proceedings of the second workshop on Automated formal methods - AFM '07* (pp. 1–10). New York, NY, USA: ACM Press. <https://doi.org/10.1145/1345169.1345170>
- Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., & Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of Clinical Epidemiology, 110*, 12–22. <https://doi.org/10.1016/j.jclinepi.2019.02.004>
- Cisco VNI. (2018). *Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper*.
- Cockburn, A. (2005). *Crystal Clear: A Human-Powered Methodology for Small Teams: A Human-Powered Methodology for Small Teams*. Addison-Wesley.
- Coronel, C., & Morris, S. (2019). *Database Systems: Design, Implementation, & Management* (13th ed). Mason, OH, United States: Cengage Learning. Retrieved from <https://www.cengage.co.uk/books/9781337627900/>
- Creately - Diagram Maker - Online Diagram Software. (2017). Retrieved May 26, 2018, from <https://creately.com/>
- Currim, S. (2008). Towards improving conceptual modeling: an examination of common errors and their underlying reasons (pp. 1–229). The University of Arizona.
- Date, C. J. . (2012). *Database Design and Relational Theory: Normal Forms and All That Jazz*. “O’Reilly Media, Inc.”
- DB-Engines. (2019a). DB-Engines Ranking - popularity ranking of database management systems. Retrieved August 27, 2019, from <https://db-engines.com/en/ranking>
- DB-Engines. (2019b). DB-Engines Ranking per database model category. Retrieved August 27, 2019, from https://db-engines.com/en/ranking_categories
- Delen, D. (2010). A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems, 49*(4), 498–506. <https://doi.org/10.1016/J.DSS.2010.06.003>
- Delen, D., Sharda, R., & Kumar, P. (2007). Movie forecast Guru: A Web-based DSS for Hollywood managers. *Decision Support Systems, 43*(4), 1151–1170. <https://doi.org/10.1016/J.DSS.2005.07.005>
- Delen, D., Walker, G., & Kadam, A. (2005). Predicting breast cancer survivability: a comparison of three data mining methods. *Artificial Intelligence in Medicine, 34*(2), 113–127. <https://doi.org/10.1016/J.ARTMED.2004.07.002>
- Derrac, J., Cornelis, C., García, S., & Herrera, F. (2012). Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. *Information Sciences, 186*(1), 73–92. <https://doi.org/10.1016/j.ins.2011.09.027>
- Dey, D., Storey, V. C., & Barron, T. M. (1999). Improving Database Design through the Analysis of Relationships. *ACM Transactions on Database Systems, 24*(4), 453–486. <https://doi.org/10.1145/331983.331984>
- Dimitrieski, V., Čeliković, M., Aleksic, S., Ristić, S., & Luković, I. (2014). Extended Entity-Relationship Approach in a Multi-Paradigm Information System Modeling Tool (pp. 1611–1620). <https://doi.org/10.15439/2014F239>
- Draw.io. (2017). Retrieved May 26, 2018, from <https://www.draw.io/>
- Drazena, G., & Ivica, C. (2017). *Bridging Relational and NoSQL Databases*. Hershey, PA,

- USA: IGI Global.
- Dustin, E., & Elfriede. (2002). *Effective software testing : 50 specific ways to improve your testing* (1° ed). Boston, MA, USA: Addison-Wesley.
- Eick, C. F., Lockemann, P. C., Eick, C. F., & Lockemann, P. C. (1985). Acquisition of terminological knowledge using database design techniques. *ACM SIGMOD Record*, 14(4), 84–94. <https://doi.org/10.1145/971699.318905>
- Eick, C. F., & Raupp, T. (1991). Towards a formal semantics and inference rules for conceptual data models. *Data & Knowledge Engineering*, 6(4), 297–317. [https://doi.org/10.1016/0169-023X\(91\)90036-W](https://doi.org/10.1016/0169-023X(91)90036-W)
- Elmadani, M., Mitrovic, A., Weerasinghe, A., & Neshatian, K. (2015). Investigating student interactions with tutorial dialogues in EER-Tutor. *Research and Practice in Technology Enhanced Learning*, 10(1), 16. <https://doi.org/10.1186/s41039-015-0013-1>
- Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems* (7th ed.). Pearson.
- ERDPlus. (2017). Retrieved May 26, 2018, from <https://erdplus.com/#/>
- Erinle, B. (2013). *Performance testing with JMeter 2.9*. Birmingham, UL: Packt Publishing.
- Everett, G. D., & McLeod, R. (2007). *Software Testing: Testing Across the Entire Software Development Life Cycle*. Wiley - IEEE John Wiley & Sons.
- Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(3), 379–383. <https://doi.org/10.3758/BF03195514>
- Finkelstein, C. (1981a, June). Information Engineering Part 4: Data Analysis And Data Base Design. *Computerworld*, Vol. 15, No. 22, 124. Retrieved from https://books.google.pt/books?id=aMrnCDJzb9MC&pg=RA1-PA1&redir_esc=y#v=onepage&q&f=false
- Finkelstein, C. (1981b, June). Information Engineering Part 5: Deriving Procedures from a Data Model. *Computerworld*, Vol. 15, No. 23, 104. Retrieved from https://books.google.pt/books?id=Ux9iw6tMs6MC&pg=PA32&redir_esc=y#v=onepage&q&f=false
- Finkelstein, C. (1998). Information Engineering Methodology. In *Handbook on architectures of information systems* (pp. 459–483). Springer. Retrieved from https://link.springer.com/content/pdf/10.1007/3-540-26661-5_19.pdf
- Finkelstein, C. (2006). *Enterprise Architecture for Integration: Rapid Delivery Methods and Technologies* (1 edition). Norwood, MA, USA: Artech House.
- Fleuret, F. (2004). *Fast Binary Feature Selection with Conditional Mutual Information*. *Journal of Machine Learning Research* (Vol. 5). Retrieved from <http://diwww.epfl.ch/>
- Genero, M., Poels, G., & Piattini, M. (2008). Defining and validating metrics for assessing the understandability of entity–relationship diagrams. *Data & Knowledge Engineering*, 64(3), 534–557. <https://doi.org/10.1016/j.datak.2007.09.011>
- Gibert, J., Valveny, E., & Bunke, H. (2012). Feature selection on node statistics based embedding of graphs. *Pattern Recognition Letters*, 33(15), 1980–1990. <https://doi.org/10.1016/j.patrec.2012.03.017>
- Godo, L., Mántaras, R. L. de, Sierra, C., & Verdager, A. (1989). MILORD: The architecture and the management of linguistically expressed uncertainty. *International Journal of Intelligent Systems*, 4(4), 471–501. <https://doi.org/10.1002/int.4550040405>
- Goldstein, C. R., & Storey, C. V. (1989). Some Findings on the Intuitivness of Entity-

- Relationship Constructs. In *Eight International Conference on Entity-Relationship Approach to Database Design and Querying* (pp. 9–23). Amsterdam. Retrieved from <https://dl.acm.org/citation.cfm?id=726856>
- González, M. de los A., & Suthers, D. D. (2000). A Coached Collaborative Learning Environment for Entity-Relationship Modeling. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *ITS: International Conference on Intelligent Tutoring Systems* (pp. 324–333). Montréal, Canada: Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45108-0_36
- Gordon C. Everest. (1976). Basic Data Structure Models Explained with a Common Example. In *Fifth Texas Conference on Computing Systems* (pp. 39–46). Austin, TX: IEEE Computer Society Publications Office.
- Green David, & Pearson J. Michael. (2016). Development of a Web Site Usability Instrument Based on ISO 9241-11. *Journal of Computer Information Systems*, 47(1), 66–72. <https://doi.org/10.1080/08874417.2006.11645940>
- Guimarães, W. W., Pinto, C. L. N., Nobre, C. N., & Zárata, L. E. (2018). Evaluation of Inductive and Transductive Inference in the context of Translation Initiation Site. <https://doi.org/10.1145/3167132.3167368>
- Hall, L., & Gordon, A. (1998). A virtual learning environment for entity relationship modelling. In *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education - SIGCSE '98* (Vol. 30, pp. 345–349). New York, USA: ACM Press. <https://doi.org/10.1145/273133.274327>
- Hambling, B., Morgan, P., Samaroo, A., Thompson, G., & Williams, P. (2010). *Software Testing: An ISTQB-ISEB Foundation Guide*. (B. Hambling, Ed.) (2nd ed.). British Comp Society Series, The Chartered Institute.
- Hartmann, S., & Link, S. (2007). English Sentence Structures and EER Modeling. In *APCCM '07 Proceedings of the fourth Asia-Pacific conference on Conceptual modelling - Volume 67* (pp. 27–35). Ballarat, Australia: Australian Computer Society, Inc. Retrieved from <http://dl.acm.org/citation.cfm?id=1274453.1274460>
- Hevner, A., & Chatterjee, S. (2010). *Design Science Research in Information Systems* (1st ed.). Springer. https://doi.org/10.1007/978-1-4419-5653-8_2
- Higgins, C. A., & Bligh, B. (2006). Formative computer based assessment in diagram based domains. *ACM SIGCSE Bulletin*, 38(3), 98. <https://doi.org/10.1145/1140123.1140152>
- Higgins, C., Symeonidis, P., & Tsintsifas, A. (2002). Diagram-based CBA using DATsys and CourseMaster. In *International Conference on Computers in Education, 2002. Proceedings.* (Vol. 1, pp. 167–172). IEEE Comput. Soc. <https://doi.org/10.1109/CIE.2002.1185893>
- Hirao, T., Okumura, M., Yasuda, N., & Isozaki, H. (2007). Supervised automatic evaluation for summarization with voted regression model. *Information Processing & Management*, 43(6), 1521–1535. <https://doi.org/10.1016/j.ipm.2007.01.012>
- Hogenboom, A., Frasinca, F., De Jong, F., & Kaymak, U. (2015). Using rhetorical structure in sentiment analysis. *Communications of the ACM*, 58(7), 69–77. <https://doi.org/10.1145/2699418>
- Hornstein, H. A. (2017). Student evaluations of teaching are an inadequate assessment tool for evaluating faculty performance. *Cogent Education*, 4(1). <https://doi.org/10.1080/2331186X.2017.1304016>
- Hosmer, D. W., & Lemeshow, S. (2005). Assessing the Fit of the Model. In *Applied Logistic*

- Regression* (Second Edition, pp. 143–202). Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/0471722146.ch5>
- Hussain, T. (2016). Teaching Entity-Relationship Models Effectively. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 264–269). IEEE. <https://doi.org/10.1109/CSCI.2016.0058>
- IBM Tivoli. (2012). Performance Test Best Practices With Rational Performance Tester. Retrieved January 10, 2019, from https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=a9ba1efe-b731-4317-9724-a181d6155e3a#fullpageWidgetId=W5f281fe58c09_49c7_9fa4_e094f86b7e98&file=b3e1526b-8981-4e42-826d-d8eadc569a13
- Ibriwesh, I., Ho, S. B., Chai, I., & Tan, C. H. (2017). A Controlled Experiment on Comparison of Data Perspectives for Software Requirements Documentation. *Arabian Journal for Science and Engineering*, 42(8), 3175–3189. <https://doi.org/10.1007/s13369-017-2425-2>
- IEEE Computer Society. (1990). IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*, 1–84. <https://doi.org/10.1109/IEEESTD.1990.101064>
- IEEE Computer Society. (2008). 829-2008 - IEEE Standard for Software and System Test. *Fredericksburg, VA, USA: IEEE Computer Society*.
- IEEE Computer Society. (2017). ISO/IEC/IEEE 26512 : 2017(E): ISO/IEC/IEEE International Standard - Systems and software engineering - Requirements for acquirers and suppliers of information for users. *ISO/IEC/IEEE 26512 First Edition, 2011-06-01*, 1–47. Retrieved from <https://ieeexplore.ieee.org/document/8288807>
- Illowsky, B., & Dean, S. (2013). *Introductory Statistics*. Rice University - OpenStax College. Retrieved from <https://openstax.org/details/books/introductory-statistics>
- ISO/IEC/IEEE. (2012). ISO/IEC/IEEE International Standard - Information technology -- Modeling Languages -- Part 2: Syntax and Semantics for IDEF1X97 (IDEFObject). *ISO/IEC/IEEE 31320-2:2012(E)*, 1–320. <https://doi.org/10.1109/IEEESTD.2012.6357338>
- ISO/IEC/IEEE. (2013). *ISO/IEC/IEEE 29119-1:2013 Software and systems engineering Software testing Part 1: Concepts and definitions. (ISO)*.
- ISO/IEC. (2008). ISO/IEC 25012:2008, Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model. Retrieved October 14, 2019, from <https://www.iso.org/obp/ui/#iso:std:iso-iec:25012:ed-1:v1:en>
- ISO. (2018). ISO 9241-11:2018(en), Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. Retrieved August 15, 2019, from <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en>
- Jaiswal, A., & Sharma, M. (2013). Expert Webest Tool: A Web Based Application, Estimate the Cost and Risk of Software Project Using Function Points (pp. 77–86). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31552-7_9
- Jajodia, S., Ng, P. A., & Springsteel, F. N. (1983). The Problem of Equivalence for Entity-Relationship Diagrams. *IEEE Transactions on Software Engineering*, SE-9(5), 617–630. <https://doi.org/10.1109/TSE.1983.235262>
- JGraph - JavaScript Diagramming. (2017). Retrieved May 26, 2018, from <https://www.jgraph.com/>
- Jie, B., Zhang, D., Cheng, B., & Shen, D. (2013). Manifold regularized multi-task feature

- selection for multi-modality classification in Alzheimer's disease. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8149 LNCS, pp. 275–283). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-40811-3_35
- Juszczak, P., Tax, D. M. J., & Duin, R. P. W. (2002). Feature scaling in support vector data description. In *Proc. ASCI* (p. 8). Citeseer.
- Kamišali'c, A. K., Heričko, M., Welzer, T., & Turkanovi'c, M. (2018). Experimental Study on the Effectiveness of a Teaching Approach Using Barker or Bachman Notation for Conceptual Database Design. *Computer Science and Information Systems*, 15(2), 421–448. <https://doi.org/10.2298/CSIS180322008K>
- Kazi, Z., Radulović, B., Berković, I., & Kazi, L. (2017). Ontology-based reasoning for entity-relationship data model semantic evaluation. *Tehnicki Vjesnik - Technical Gazette*, 24(Supplement 1), 39–47. <https://doi.org/10.17559/TV-20140711141546>
- Kesh, S. (1995). Evaluating the quality of entity relationship models. *Information and Software Technology*, 37(12), 681–689. [https://doi.org/10.1016/0950-5849\(96\)81745-9](https://doi.org/10.1016/0950-5849(96)81745-9)
- Khoshgoftaar, T., Dittman, D., Wald, R., & Fazelpour, A. (2012). First order statistics based feature selection: A diverse and powerful family of feature selection techniques. In *Proceedings - 2012 11th International Conference on Machine Learning and Applications, ICMLA 2012* (Vol. 2, pp. 151–157). <https://doi.org/10.1109/ICMLA.2012.192>
- Kiang, M. Y., & Y., M. (2003). A comparative assessment of classification methods. *Decision Support Systems*, 35(4), 441–454. [https://doi.org/10.1016/S0167-9236\(02\)00110-0](https://doi.org/10.1016/S0167-9236(02)00110-0)
- Kim, N., Lee, S., & Moon, S. (2008). Formalized Entity Extraction Methodology for Changeable Business Requirements. *Journal of Information Science and Engineering*, 24(3), 649–671.
- Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum - making the most of both (Enterprise Software Development)*. (D. Plesa, Ed.). C4Media, Inc.
- Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2 IJCAI'95* (pp. 1137–1143). Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc.
- Kortum, P. T., & Bangor, A. (2013). Usability Ratings for Everyday Products Measured With the System Usability Scale. *International Journal of Human-Computer Interaction*, 29(2), 67–76. <https://doi.org/10.1080/10447318.2012.681221>
- Kroenke, D. M., Auer, D. J., Vandenberg, S. L., & Yoder, R. C. (2018). *Database Processing: Fundamentals, Design, and Implementation* (15 ed). Pearson.
- Kroenke, D. M., Auer, D. J., Vandenberg, S. L., & Yoder, R. C. (2019). *Database concepts* (9 ed). Pearson.
- Kruchten, P. (2004). *The Rational Unified Process: An Introduction*. Addison-Wesley.
- Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling. Applied Predictive Modeling*. Springer New York. <https://doi.org/10.1007/978-1-4614-6849-3>
- Kumar, S., & Chong, I. (2018). Correlation Analysis to Identify the Effective Data in Machine Learning: Prediction of Depressive Disorder and Emotion States. *International Journal of Environmental Research and Public Health*, 15(12). <https://doi.org/10.3390/ijerph15122907>

- Kurose, J., & Marzullo, K. (2016). *THE FEDERAL BIG DATA RESEARCH AND DEVELOPMENT STRATEGIC PLAN THE NETWORKING AND INFORMATION TECHNOLOGY RESEARCH AND DEVELOPMENT PROGRAM About the NITRD Big Data Senior Steering Group*. Retrieved from <https://www.nitrd.gov/pubs/bigdatardstrategicplan.pdf>
- Lane, M. C., Xue, B., Liu, I., & Zhang, M. (2014). Gaussian based particle swarm optimisation and statistical clustering for feature selection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8600, pp. 133–144). Springer Verlag. https://doi.org/10.1007/978-3-662-44320-0_12
- Lewis, J. R. (2018). The System Usability Scale: Past, Present, and Future. *International Journal of Human-Computer Interaction*, 34(7), 577–590. <https://doi.org/10.1080/10447318.2018.1455307>
- Lewis, J. R., & Sauro, J. (2009). The Factor Structure of the System Usability Scale. In M. Kurosu (Ed.), *Human Centered Design* (pp. 94–103). Berlin, Heidelberg: Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02806-9_12
- Li, B., Zheng, C. H., & Huang, D. S. (2008). Locally linear discriminant embedding: An efficient method for face recognition. *Pattern Recognition*, 41(12), 3813–3821. <https://doi.org/10.1016/j.patcog.2008.05.027>
- Li, H., Li, C. J., Wu, X. J., & Sun, J. (2014). Statistics-based wrapper for feature selection: An implementation on financial distress identification with support vector machine. *Applied Soft Computing Journal*, 19, 57–67. <https://doi.org/10.1016/j.asoc.2014.01.018>
- Liao, C., & Palvia, P. C. (2000). The impact of data models and task complexity on end-user performance: An experimental investigation. *International Journal of Human Computer Studies*, 52(5), 831–845. <https://doi.org/10.1006/ijhc.1999.0358>
- Lino, A. D. P. (2007). *LABSQL: laboratório de ensino de SQL*. Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica. Master's Thesis, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Pará. Retrieved from <http://repositorio.ufpa.br/jspui/handle/2011/1744>
- Lino, A., Rocha, Á., Macedo, L., & Sizo, A. (2019). Application of clustering-based decision tree approach in SQL query error database. *Future Generation Computer Systems*, 93, 392–406. <https://doi.org/10.1016/j.future.2018.10.038>
- Lino, Adriano, Rocha, Á., & Sizo, A. (2016). Virtual teaching and learning environments: Automatic evaluation with symbolic regression. *Journal of Intelligent & Fuzzy Systems*, 31(4), 2061–2072. <https://doi.org/10.3233/JIFS-169045>
- Lucassen, G., Robeer, M., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2017). Extracting conceptual models from user stories with Visual Narrator. *Requirements Engineering*, 22(3), 339–358. <https://doi.org/10.1007/s00766-017-0270-1>
- Lucidchart - Online Diagram Software & Visual Solution | Lucidchart. (2017). Retrieved May 26, 2018, from <https://www.lucidchart.com/>
- Mak, R. (2009). *Writing Compilers and Interpreters: A Modern Software Engineering Approach* (3 ed). John Wiley & Sons.
- Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press.
- Markow, W., Braganza, S., Taska, B., Miller, S. M., & Hughes, D. (2017). *THE QUANT CRUNCH HOW THE DEMAND FOR DATA SCIENCE SKILLS IS DISRUPTING THE*

- JOB MARKET.* Boston, MA, USA. Retrieved from <https://www.ibm.com/downloads/cas/3RL3VXGA>
- Martin, J., & Finkelstein, C. (1981). *Information engineering* (2 ed). Carnforth, UK: Savant Institute.
- Meier, J. D., Farre, C., Bansode, P., Barber, S., & Rea, D. (2007a). Chapter 7 – Managing the Performance Test Cycle in a Regulated (CMMI) Environment. Retrieved May 26, 2018, from [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/bb924362\(v%3Dpandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/bb924362(v%3Dpandp.10))
- Meier, J. D., Farre, C., Bansode, P., Barber, S., & Rea, D. (2007b). Performance Testing Guidance for Web Applications, Chapter 15 – Key Mathematic Principles for Performance Testers. Retrieved January 10, 2019, from [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/bb924370\(v%3Dpandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/bb924370(v%3Dpandp.10))
- Mellor, S. J., Scott, K., Uhl, A., & Weise, D. (2002). Model-Driven Architecture (pp. 290–297). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-46105-1_33
- Melo, B. M., Lino, A. D. P., Figueira, A. S., Silva, A. G., Lobato, F. R. L., Morais, A. S., & Favero, E. L. (2013). Assessment module automatic SQL code with feedback in the form of tips in the virtual learning environment LabSQL. In *2013 8th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1–7). Lisboa, Portugal: IEEE.
- MindFusion - JavaScript Diagram Library Drawing Library For Flowcharts. (2017). Retrieved May 26, 2018, from <https://mindfusion.eu/javascript-diagram.html>
- Mitrovic, A. (2012). Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 22(1–2), 39–72. <https://doi.org/10.1007/s11257-011-9105-9>
- Mitrovic, A., & Suraweera, P. (2016). Teaching Database Design with Constraint-Based Tutors. *International Journal of Artificial Intelligence in Education*, 26(1), 448–456. <https://doi.org/10.1007/s40593-015-0084-6>
- Moody, D. (2003). Measuring the Quality of Data Models: An Empirical Evaluation of the Use of Quality Metrics in Practice. In *ECIS 2003 Proceedings* (p. 18).
- Moody, D. L. (1998). Metrics for Evaluating the Quality of Entity Relationship Models (pp. 211–225). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-49524-6_18
- Moody, D. L. (2004). Cognitive load effects on end user understanding of conceptual models: An experimental analysis. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3255, 129–143. https://doi.org/10.1007/978-3-540-30204-9_9
- Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3), 243–276. <https://doi.org/10.1016/j.datak.2004.12.005>
- Moody, D. L., & Shanks, G. G. (1994). What makes a good data model? Evaluating the quality of entity relationship models. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 881 LNCS, pp. 94–111). Springer Verlag. https://doi.org/10.1007/3-540-58786-1_75
- Moody, D. L., & Shanks, G. G. (2003). Improving the quality of data models: empirical validation of a quality management framework. *Information Systems*, 28(6), 619–650. [https://doi.org/10.1016/S0306-4379\(02\)00043-1](https://doi.org/10.1016/S0306-4379(02)00043-1)
- Moody, D. L., Shanks, G. G., & Darke, P. (1998). Improving the Quality of Entity Relationship Models—Experience in Research and Practice (pp. 255–276). Springer, Berlin,

- Heidelberg. https://doi.org/10.1007/978-3-540-49524-6_21
- Morrison, P., Moye, D., Pandita, R., & Williams, L. (2018). Mapping the field of software life cycle security metrics. *Information and Software Technology*, 102, 146–159. <https://doi.org/10.1016/J.INFSOF.2018.05.011>
- mxGraph. (2017). Retrieved May 26, 2018, from <https://www.jgraph.com/mxchangelog.html>
- Myers, G. J., Sandler, C., & Badgett, T. (2012). *The art of software testing*. John Wiley & Sons.
- MySQL. (2018). MySQL :: MySQL 8.0 Reference Manual. Retrieved August 21, 2018, from <https://dev.mysql.com/doc/refman/8.0/en/>
- Nazir, F., Butt, W. H., Anwar, M. W., & Khan Khattak, M. A. (2017). The applications of natural language processing (NLP) for software requirement engineering - A systematic literature review. In *Lecture Notes in Electrical Engineering* (Vol. 424, pp. 485–493). Springer Verlag. https://doi.org/10.1007/978-981-10-4154-9_56
- Nguyen Thai Nghe, Janecek, P., & Haddawy, P. (2007). A comparative analysis of techniques for predicting academic performance. In *2007 37th annual frontiers in education conference - global engineering: knowledge without borders, opportunities without passports* (pp. T2G-7-T2G-12). IEEE. <https://doi.org/10.1109/FIE.2007.4417993>
- Nielsen, J. (1993). *Usability engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from <https://dl.acm.org/citation.cfm?id=529793>
- Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93* (pp. 206–213). New York, USA: ACM Press. <https://doi.org/10.1145/169059.169166>
- NIST. (1993). FIPS Publication 184: Integration Definition of Information Modeling (IDEF1X). *National Institute of Standards and Technology*.
- OctoPerf. (2019). What is OctoPerf? Retrieved August 15, 2019, from <https://doc.octoperf.com/>
- Ognjenovic, P. (2017). Experimental verification of the effectiveness of learning and teaching using intelligent tutoring system in secondary school education. In *2017 25th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2017*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.23919/SOFTCOM.2017.8115537>
- Ohlsson, S. (1994). Constraint-Based Student Modeling. In *Student Modelling: The Key to Individualized Knowledge-Based Instruction* (pp. 167–189). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-03037-0_7
- Ondřej Žára. (2012). WWW SQL Designer - Online tool for designing relational database schemas. Retrieved May 26, 2018, from <http://ondras.zarovi.cz/#projects>
- Oppel, A. J. (2011). *Databases DeMystiFieD* (2 edition). McGraw-Hill.
- Oppenheim, B. W. (2011). *Lean for Systems Engineering with Lean Enablers for Systems Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9781118063996>
- Park, H., & Choi, J. (2012). V-model. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (p. 453). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2207676.2207739>
- Parlamento Europeu. (2016). REGULAMENTO (UE) 2016/679 DO PARLAMENTO

- EUROPEU E DO CONSELHO de 27 de abril de 2016 relativo. *Jornal Oficial Da União Europeia*, 88.
- Parsons, J., & Saunders, C. (2004). Cognitive heuristics in software engineering: Applying and extending anchoring and adjustment to artifact reuse. *IEEE Transactions on Software Engineering*, 30(12), 873–888. <https://doi.org/10.1109/TSE.2004.94>
- Peckham, J., & Maryanski, F. (1988). Semantic Data Models. *ACM Computing Surveys (CSUR)*, 20(3), 153–189. <https://doi.org/10.1145/62061.62062>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Perrot, M. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(1), 2825–2830. Retrieved from <http://scikit-learn.sourceforge.net>.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-122240302>
- Pendegraft, N. (2012). A Teaching Note on Demonstrating the Ambiguity of ERDs with Respect to Referential Integrity. In *Proceedings of the 2012 InSITE Conference* (pp. 151–156). Informing Science Institute. <https://doi.org/10.28945/1645>
- Peng, H., Long, F., & Ding, C. (2005). Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), 1226–1238. <https://doi.org/10.1109/TPAMI.2005.159>
- Piprani, B., & Ernst, D. (2008). A Model for Data Quality Assessment (pp. 750–759). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88875-8_99
- Poels, G., & Dedene, G. (2000). Distance-based software measurement: necessary and sufficient properties for software measures. *Information and Software Technology*, 42(1), 35–46. [https://doi.org/10.1016/S0950-5849\(99\)00053-1](https://doi.org/10.1016/S0950-5849(99)00053-1)
- Pons, A. P., Polak, P., & Stutz, J. (2005). Evaluating the Teaching Effectiveness of Various Data Modeling Notations. *Journal of Computer Information Systems*, 46(2), 78–84. <https://doi.org/10.1080/08874417.2006.11645886>
- Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97, 205–227. <https://doi.org/10.1016/j.eswa.2017.12.020>
- Pradipta, K. B., Aneesh, K. K. V., Dipankar, S., & Vaidyanathan, S. (2017). Software based collection of performance metrics for allocation adjustment of virtual resources. Retrieved from <https://patents.google.com/patent/US20180203738A1/en>
- Prakosa Budi. (2015). Laravel Database Designer 0.1. Retrieved May 26, 2018, from <http://biodesignrealworld.github.io/LaravelDatabaseDesigner/>
- Pressman, R. S., & Maxim, B. R. (2015). *Software Engineering: A Practitioner's Approach - Eighth edition* (Eighth edition). New York, NY: McGraw-Hill Education.
- Puja, I., Posic, P., & Jaksic, D. (2019). Overview and Comparison of Several Relational Database Modelling Metodologies and Notations. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1641–1646). Opatija, Croatia: IEEE. <https://doi.org/10.23919/mipro.2019.8756667>
- QuickDBD - Draw database diagrams by typing. (2017). Retrieved May 26, 2018, from <https://www.quickdatabasediagrams.com/>

- Ravi Kumar, T., Srinivasa Rao, T., & Bathini, S. (2019). A Predictive Approach to Estimate Software Defects Density Using Weighted Artificial Neural Networks for the Given Software Metrics (pp. 449–457). Springer, Singapore. https://doi.org/10.1007/978-981-13-1927-3_48
- Reinsel, D., Gantz, J., & Rydning, J. (2018). *The Digitization of the World From Edge to Core*.
- Rempel, G., & Garret. (2015). Defining Standards for Web Page Performance in Business Applications. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering - ICPE '15* (pp. 245–252). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2668930.2688056>
- Rice, M. E., & Harris, G. T. (2005). Comparing Effect Sizes in Follow-Up Studies: ROC Area, Cohen's d, and r. *Law and Human Behavior*, 29(5). <https://doi.org/10.1007/s10979-005-6832-7>
- Riihiaho, S. (2000). *Experiences with usability evaluation methods*. Helsinki University of Technology.
- Royce, W. W. (1987). Managing the Development of Large Software Systems: Concepts and Techniques. In *Proceedings of the 9th International Conference on Software Engineering* (pp. 328–338). Monterey, California, USA: IEEE Computer Society Press.
- Russell, S., & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach* (3ed ed.). Upper Saddle River, NJ, USA: Prentice Hall Press.
- Saiedian, H. (1997). An evaluation of extended entity-relationship model. *Information and Software Technology*, 39(7), 449–462. [https://doi.org/10.1016/S0950-5849\(97\)00002-5](https://doi.org/10.1016/S0950-5849(97)00002-5)
- SCHWAB, K. (2018). *The Future of Jobs Report 2018 Insight Report Centre for the New Economy and Society*. Cologny/Geneva, Switzerland. <https://doi.org/978-1-944835-18-7>
- Schwaber, K., & Sutherland, J. (2012). *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*. (K. Schwaber & J. Sutherland, Eds.). Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9781119203278>
- Sebesta, R. W. (2015). *Concepts of programming languages* (11ed ed.). Pearson.
- Shafique, U., & Qaiser, H. (2014). A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, 12(1), 217–222. <https://doi.org/10.1.1.686.840>
- Shahiri, A. M., Husain, W., & Rashid, N. A. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. *Procedia Computer Science*, 72, 414–422. <https://doi.org/10.1016/j.procs.2015.12.157>
- Sharda, R., & Delen, D. (2006). Predicting box-office success of motion pictures with neural networks. *Expert Systems with Applications*, 30(2), 243–254. <https://doi.org/10.1016/J.ESWA.2005.07.018>
- Shen, L., & Bai, L. (2006). Information Theory for Gabor Feature Selection for Face Recognition. *EURASIP Journal on Applied Signal Processing*, 30274, 1–11. <https://doi.org/10.1155/ASP/2006/30274>
- Shepperd, M., & Ince, D. (1991). Algebraic validation of software metrics (pp. 343–363). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3540547428_57
- Shoval, P., & Shiran, S. (1997). Entity-relationship and object-oriented data modeling - An experimental comparison of design quality. *Data and Knowledge Engineering*, 21(3), 297–315. [https://doi.org/10.1016/S0169-023X\(97\)88935-5](https://doi.org/10.1016/S0169-023X(97)88935-5)

- Shu, W., & Shen, H. (2014). Incremental feature selection based on rough set in dynamic incomplete data. *Pattern Recognition*, 47(12), 3890–3906. <https://doi.org/10.1016/j.patcog.2014.06.002>
- Silberschatz, A., Korth, H., & Sudarshan, S. (2010). *Entity relationship model. Database System Concepts, sixth ed.*, McGraw-Hill. McGraw-Hill.
- Silva, R. O., Machado, G. B. G., Viana, G. B., & Santos Silva, J. (2016). O Processo de Teste de Software. *Tecnologias Em Projeção*, 7(2), 1–10.
- Simsion, G. (2014). *Data Modeling Theory and Practice* (1ed ed.). Technics Publications.
- Singh, R., Sharma, V., & Varshney, M. (2009). *Design and implementation of compiler*. New Age International.
- SmartDraw - Create Flowcharts, Floor Plans, and Other Diagrams on Any Device. (2017). Retrieved May 26, 2018, from https://www.smartdraw.com/?id=104640&gclid=Cj0KCQjw6J7YBRC4ARIsAJMXXscmtUlpIsijrpIXXRLyz8u80cNMiud_DuJ5hqzF_26_HUjkhJg9PQaAs8pEALw_wcB
- Sommerville, I. (2016). *Software Engineering. Pearson Education Limited* (10ed ed.). Pearson.
- Song, I.-Y., Evans, M., & Park, U. E. K. (1995). A Comparative Analysis of Entity-Relationship Diagrams. *Journal of Computer and Software Engineering*, 3(34), 427–459.
- Song, I.-Y., Yano, K., Trujillo, J., & Lujan-Mora, S. (2011). A Taxonomic Class Modeling Methodology for Object-Oriented Analysis. In *Information Modeling Methods and Methodologies* (pp. 216–240). IGI Global. <https://doi.org/10.4018/978-1-59140-375-3.ch011>
- Song, I. Y., Zhu, Y., Ceong, H., & Thonggoom, O. (2015). Methodologies for semi-automated conceptual data modeling from requiremethodologies. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9381, pp. 18–31). Springer Verlag. https://doi.org/10.1007/978-3-319-25264-3_2
- Speelpenning, J., Daux, P., & Gallus, J. (2001). *Data Modeling and Relational Database Design: Student Guide*. (Avril Price-Budgen, Fiona Simpson, & Don Griffin, Eds.) (2ed ed.). Oracle Corporation.
- SQLDBM - SQL Server Database Modeler, Entity Relationship Diagram. (2017). Retrieved May 26, 2018, from <https://sqldb.com/en/Home/>
- Stephens Rod. (2015). *Beginning Software Engineering* (1ed ed.). Wrox. <https://doi.org/8126555378>
- Strickland, J. S. (2011). *Systems Engineering Processes and Practice* (1ed ed.). Lulu, Inc.
- Su, Y., & Yan, S. Y. (2011). *Principles of Compilers: A New Approach to Compilers Including the Algebraic Method*. Springer Science & Business Media.
- Sugumaran, V., & Storey, V. C. (2006). The role of domain ontologies in database design: An ontology management and conceptual modeling environment. *ACM Transactions on Database Systems*, 31(3), 1064–1094. <https://doi.org/10.1145/1166074.1166083>
- Suraweera, P., & Mitrovic, A. (2002). KERMIT: A Constraint-Based Tutor for Database Modeling (pp. 377–387). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-47987-2_41
- Suraweera, P., & Mitrovic, A. (2004). An Intelligent Tutoring System for Entity Relationship Modelling. *International Journal of Artificial Intelligence in Education*, 14(3), 375–417.
- Swiniarski, R. W., & Skowron, A. (2003). Rough set methods in feature selection and

- recognition. *Pattern Recognition Letters*, 24(6), 833–849. [https://doi.org/10.1016/S0167-8655\(02\)00196-4](https://doi.org/10.1016/S0167-8655(02)00196-4)
- Tavana, M., Joglekar, P., & Redmond, M. A. (2007). An automated entity–relationship clustering algorithm for conceptual database design. *Information Systems*, 32(5), 773–792. <https://doi.org/10.1016/J.IS.2006.07.001>
- Teorey, T. J., Lightstone, S. S., Nadeau, T., & Jagadish, H. V. (2011). *Database Modeling and Design: Logical Design* (5ed ed.). Elsevier.
- Teorey, T. J., Yang, D., & Fry, J. P. (1986). A logical design methodology for relational databases using the extended entity-relationship model. *ACM Computing Surveys*, 18(2), 197–222. <https://doi.org/10.1145/7474.7475>
- Thomas, P. (2013). Online Automatic Marking of Diagrams. *Systemic Practice and Action Research*, 26(4), 349–359. <https://doi.org/10.1007/s11213-012-9273-5>
- Thomas, P., Smith, N., & Waugh, K. (2008). Automatically assessing graph-based diagrams. *Learning, Media and Technology*, 33(3), 249–267. <https://doi.org/10.1080/17439880802324251>
- Thomas, P., Waugh, K., Smith, N., Thomas, P., Waugh, K., & Smith, N. (2005). Experiments in the automatic marking of ER-diagrams. *ACM SIGCSE Bulletin*, 37(3), 158. <https://doi.org/10.1145/1151954.1067490>
- Thomas, P., Waugh, K., Smith, N., Thomas, P., Waugh, K., & Smith, N. (2006). Using patterns in the automatic marking of ER-diagrams. In *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education - ITICSE '06* (Vol. 38, p. 83). New York, NY, USA: ACM Press. <https://doi.org/10.1145/1140124.1140149>
- Topi, H., & Ramesh, V. (2002). Human Factors Research on Data Modeling: A Review of Prior Research, An Extended Framework and Future Research Directions. *Journal of Database Management*, 13(2), 3–19. <https://doi.org/10.4018/jdm.2002040101>
- Tracz, W. (2002). Extreme Programming (XP). In *Encyclopedia of Software Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/0471028959.sof112>
- Tsintsifas, A. (2002). *A framework for the computer based assessment of diagram based coursework*. *Computer Science and Information Technology*. Loughborough University. Retrieved from <http://www.cs.nott.ac.uk/~cah/pdf/azt-phd.pdf>
- Vasconcelos, M., & Vasconcelos, N. (2009). Natural image statistics and low-complexity feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2), 228–244. <https://doi.org/10.1109/TPAMI.2008.77>
- Wan, X. (2019). Influence of feature scaling on convergence of gradient iterative algorithm. *Journal of Physics: Conference Series*, 1213(3), 032021. <https://doi.org/10.1088/1742-6596/1213/3/032021>
- Wang, J., Guo, K., & Wang, S. (2010). Rough set and Tabu search based feature selection for credit scoring. *Procedia Computer Science*, 1(1), 2425–2432. <https://doi.org/10.1016/j.procs.2010.04.274>
- Watson, D. (2017). *A Practical Approach to Compiler Construction*. Springer. <https://doi.org/10.1007/978-3-319-52789-5>
- Weilkiens Tim, Jesko G. Lamm, Stephan Roth, & Markus Walker. (2015). Agile Approaches. In *Model-Based System Architecture* (pp. 183–188). Hoboken, NJ, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9781119051930.ch13>

- Wieggers, K. E., & Beatty, J. (2013). *Software requirements* (3ed ed.). Pearson Education.
- Xu, Z., King, I., Lyu, M. R. T., & Jin, R. (2010). Discriminative Semi-Supervised Feature Selection Via Manifold Regularization. *IEEE Transactions on Neural Networks*, 21(7), 1033–1047. <https://doi.org/10.1109/TNN.2010.2047114>
- Yang, H. H., & Moody, J. (1999). Data Visualization and Feature Selection: New Algorithms for Nongaussian Data. In *Proceedings of the 12th International Conference on Neural Information Processing Systems* (pp. 687–693). Denver, CO: MIT Press.
- Yang, L., & Cao, L. (2016). The Effect of MySQL Workbench in Teaching Entity-Relationship Diagram (ERD) to Relational Schema Mapping. *Article in International Journal of Modern Education and Computer Science*, 7, 1–12. <https://doi.org/10.5815/ijmeecs.2016.07.01>
- Yu, K., Wu, X., Ding, W., & Pei, J. (2016). Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data*, 11(2), 1–39. <https://doi.org/10.1145/2976744>
- Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, 16(1), 39. <https://doi.org/10.1186/s41239-019-0171-0>
- Zhao, C., Sviridenko, D., Torgersen, N., Vliet, R. van der, & Nianios, R. (2017). Free Online Database Design Tool - DB Designer. Retrieved May 26, 2018, from <https://www.dbdesigner.net/>

Anexo A: Lista de livros de BD

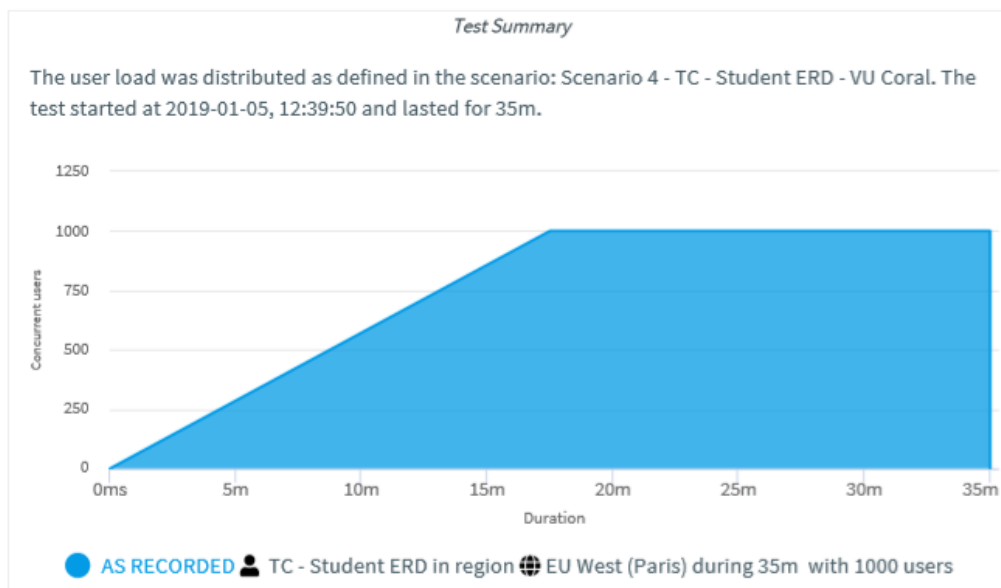
Autor	Título	Ano	ISBN	Incluso
Coronel C., Morris S., Rob P.	Database Systems. Design, Implementation, and Management 10ed	2012	1111969604	SIM
Date C.J.	Database modeling and relational theory	2012	9781449328016	SIM
Teorey T.J., Lightstone S.S., Nadeau T., Jagadish H.V.	Database modeling and design. logical design 5ed	2011	9780123820204	SIM
Elmasri Navathe	Fundamentals of Database Systems - 6ed	2010	9780136086208	SIM
Oppel A.	Databases DeMYSTiFieD 2ed	2010	0071747990	SIM
Silberschatz A., Korth H., Sudarshan S.	Database System Concepts 6ed	2010	0073523321	SIM
Liu L., Ozsu M. (eds.)	Encyclopedia of database systems	2009	9780387355443	SIM
Stephens R.	Beginning database design solutions	2009	0470385499	SIM
Teorey T.J.	Database modeling and design 3ed	2009	1558605002	SIM
Joe Celko	SQL for smarties - Advanced SQL Programming 3ed	2005	9780123693792	SIM
Teorey T.J., Lightstone S.S., Nadeau T.	Database modeling and design. logical design 4ed	2005	0126853525	SIM
Gulutzan P., Pelzer T.	SQL performance tuning	2003	0201791692	SIM
Hernandez M.J.	Database design for mere mortals 2ed	2003	0201752840	SIM
Ullman J.D., Widom J.D.	A First Course in Database Systems 2ed	2001	9780130353009	SIM
Ramakrishnan R., Gehrke J.	Database Management Systems 2ed	2000	0072465352	SIM
Ullman J.D., Widom J.D.	A first course in database systems	1997	0138613370	SIM
Teorey T.J.	Database modeling and design. the fundamental principles 2ed.	1994	1558602947	SIM
Bell C.	Expert MySQL 2ed	2012	9781430246596	NÃO
Date C.J.	SQL and relational theory. how to write accurate SQL code 2ed	2011	1449316409	NÃO
Jiawei Han Micheline Kamber Jian Pei	Data Mining Concepts and Techniques 3ed	2011	9780123814791	NÃO
Date C.J.	SQL and relational theory. how to write accurate SQL code	2009	0596523068	NÃO
Date C.J.	The Relational Database Dictionary, Extended Edition	2008	1430210419	NÃO
Jiawei Han Micheline Kamber	Data Mining Concepts and Techniques 2ed	2007	9781558609013	NÃO
Date C.J.	Date on Database. Writings 2000-2006	2006	9781590597460	NÃO
Whitehorn M., Marklyn B.	Inside Relational Databases with Examples in Access	2006	1846283949	NÃO
Date C.	Database in Depth. Relational Theory for Practitioners	2005	0596100124	NÃO
Date C.J.	Database in Depth. Relational theory for practitioners	2005	0596100124	NÃO
Celko J. Joe Celko's	Trees and Hierarchies in SQL for Smarties	2004	1558609202	NÃO
Date C.J.	Introduction to database systems 8ed.	2004	0321189566	NÃO
Date C.J.	Instructor's Manual for Introduction to Database Systems 8ed	2003	0321233905.	NÃO
Shekhar S., Chawla S.	Spatial Databases. A Tour	2003	9780130174802	NÃO
Kimball R., Ross M.	The data warehouse toolkit 2ed	2002	0471200247	NÃO
Garcia-Molina H., Ullman J.D., Widom J.D.	Database Systems. The Complete Book	2001	9780130319951	NÃO
Jan Speelpenning, Patrice Daux, Jeff Gallus	Data Modeling and Relational Database Design Oracle	2001	Student BOOK	NÃO
Garcia-Molina H., Ullman J.D., Widom J.D.	Database System Implementation	1999	0130402648	NÃO
Date C.J., Darwen H.	Foundation for object relational databases	1998	0201309785	NÃO
Lausen G., Vossen G.	Models and languages of object-oriented databases	1998	0201624311	NÃO
Stonebraker M., Brown P., Moore D.	Object-Relational DBMSs 2ed	1998	9781558604520	NÃO
Date C.J., Darwen H.	A guide to the SQL standard 4ed	1997	0201964260	NÃO
Date C.J.	Relational database writings 1991-1994	1995	0201824590	NÃO
Bancilhon F., Buneman P	Advances in database programming languages	1990	0201502577	NÃO
Date C.J.	Relational database writings 1985-1989	1990	0201508818	NÃO
Date C.J.	A guide to the SQL standard 2ed	1989	0201502097	NÃO
Alagic S.	Relational database technology	1986	038796276X	NÃO
Date C.J.	An introduction to database systems, vol.2	1985	0201144743	NÃO
Date C., Darwen H., Lorentzos N.	Temporal Data and the Relational Model	2002	1558608559	NÃO
Date C.J.	An Introduction to Database Systems. 6ed	1994	9780201824582	NÃO
Date C.J., Darwen H.	Relational database writings 1989-1991	1992	0201543036	NÃO







Anexo B: Teste de Desempenho

O teste de desempenho do LabDER foi executado em parceria com a empresa OctoPerf, e neste Anexo os resultados dos testes são apresentados no formato original dos relatórios da ferramenta para quatro cenários de teste. TC-Student ERD, TC - Student View Content, TC - Student SQL, TC - Professor Analyze Results e TC Professor ImportCSV. No final, é apresentado um glossário com a lista de terminologias utilizadas.

TC – Student ERD

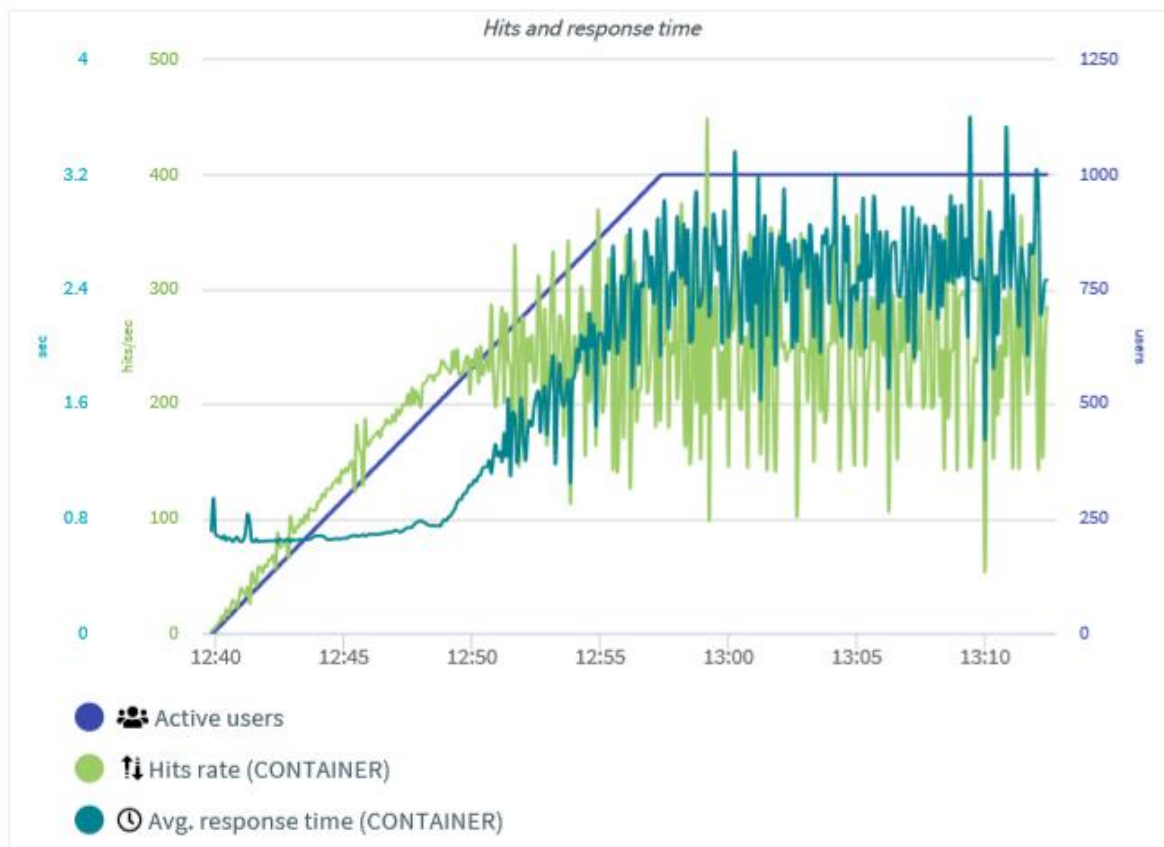
1. Test Summary



Statistics summary	
	Hits count 3136723 hits
	Errors count 50 errors
	Avg. response time (CONTAINER) 1.822 sec
	Latency standard deviation (CONTAINER) 1.13 sec
	Received Bytes rate 6.4 MB/sec
	Apdex on response time (CONTAINER) 0.562 apdex

2. Results Summary

2.1. Hit rate and response time under load



2.2. Request details

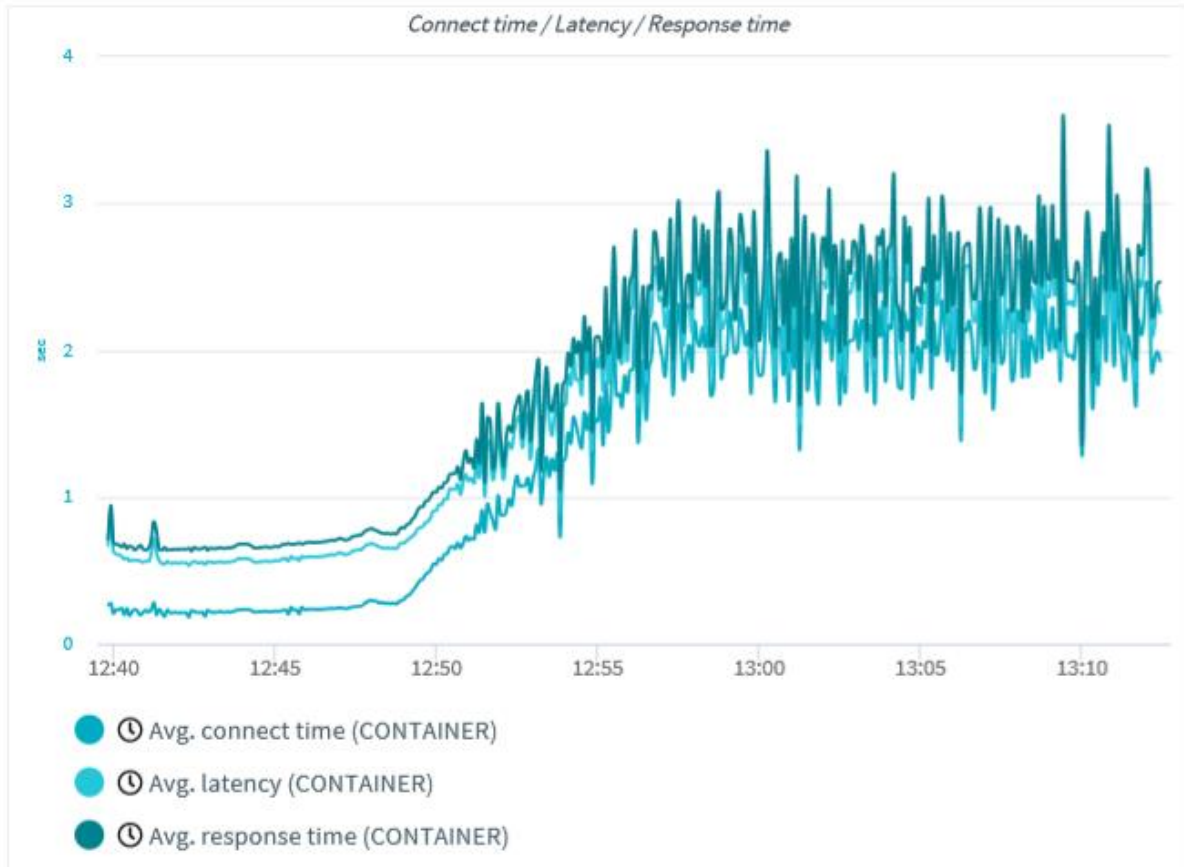
<i>Results tree</i>					
	Action	Avg. response time (sec)	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
	 AS RECORDED - T...				
	 TC - Student L...	1.744	2.638	83224	11
	 TC - Student A...	2.113	3.294	82876	16
	 TC - ERD	3.233	4.693	82441	16
	 EV - Submit E...	1.168	1.831	82354	4
	 TC - Student L...	0.848	1.339	82071	3

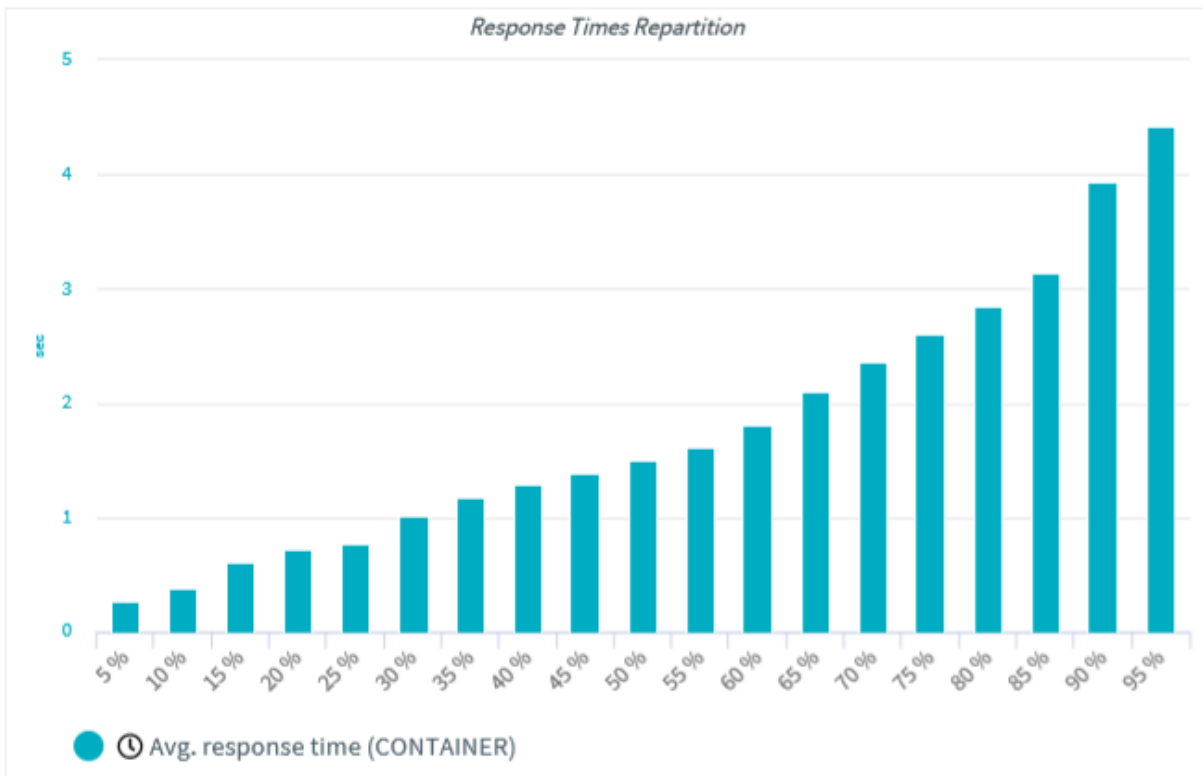
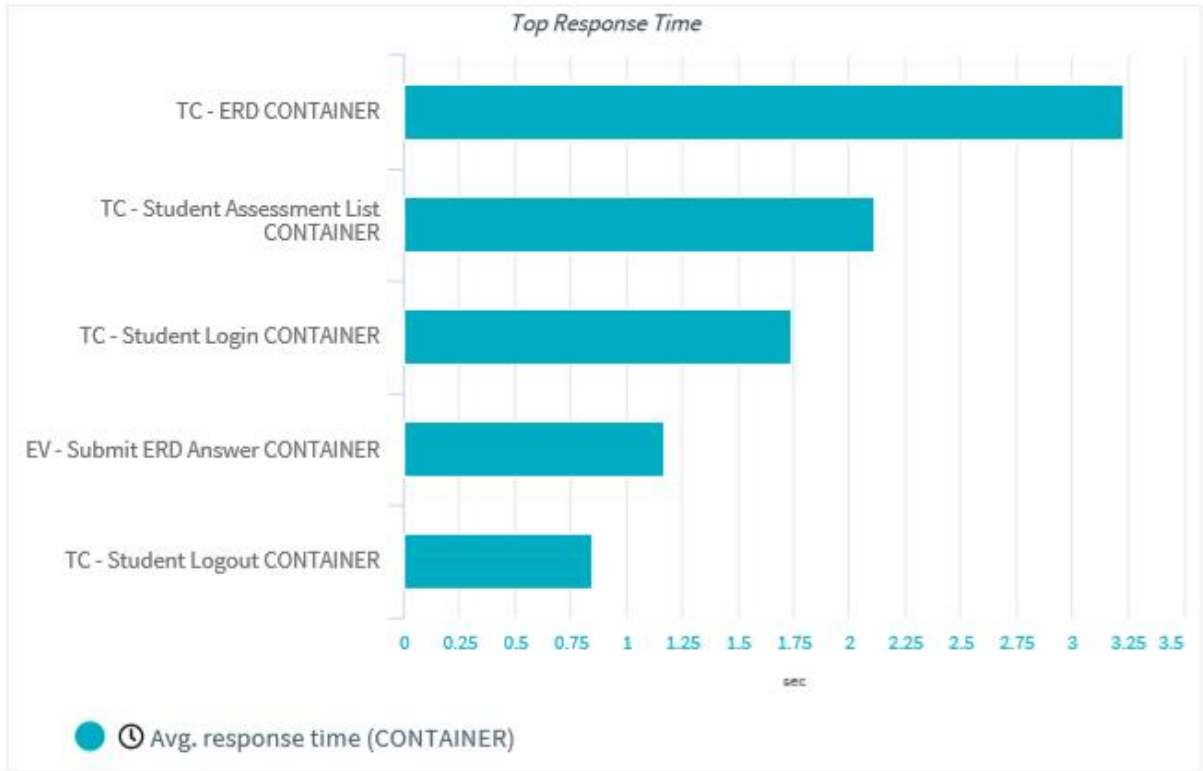
Total Items: 44 (Showing Items: 6)






-   Avg. response time
-   Response time percentile 90
-   Hits count
-   Errors count

3. Result breakdown





3.1. Response time and percentiles



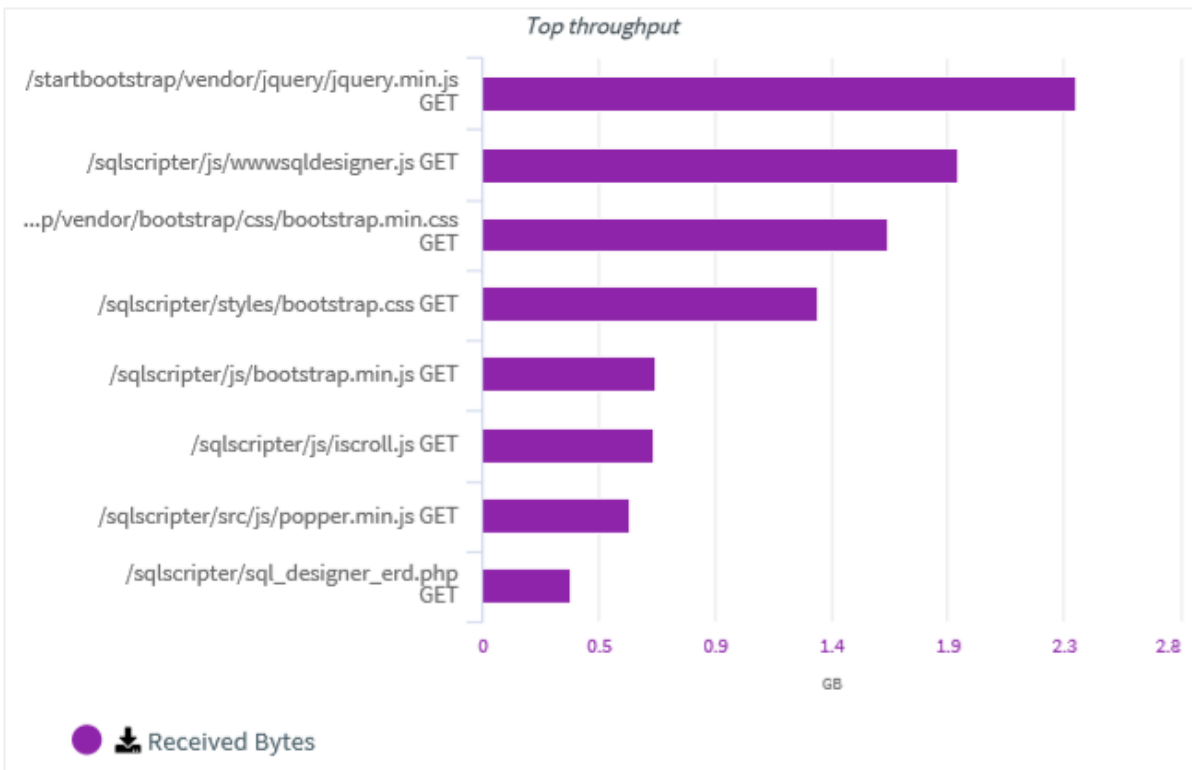
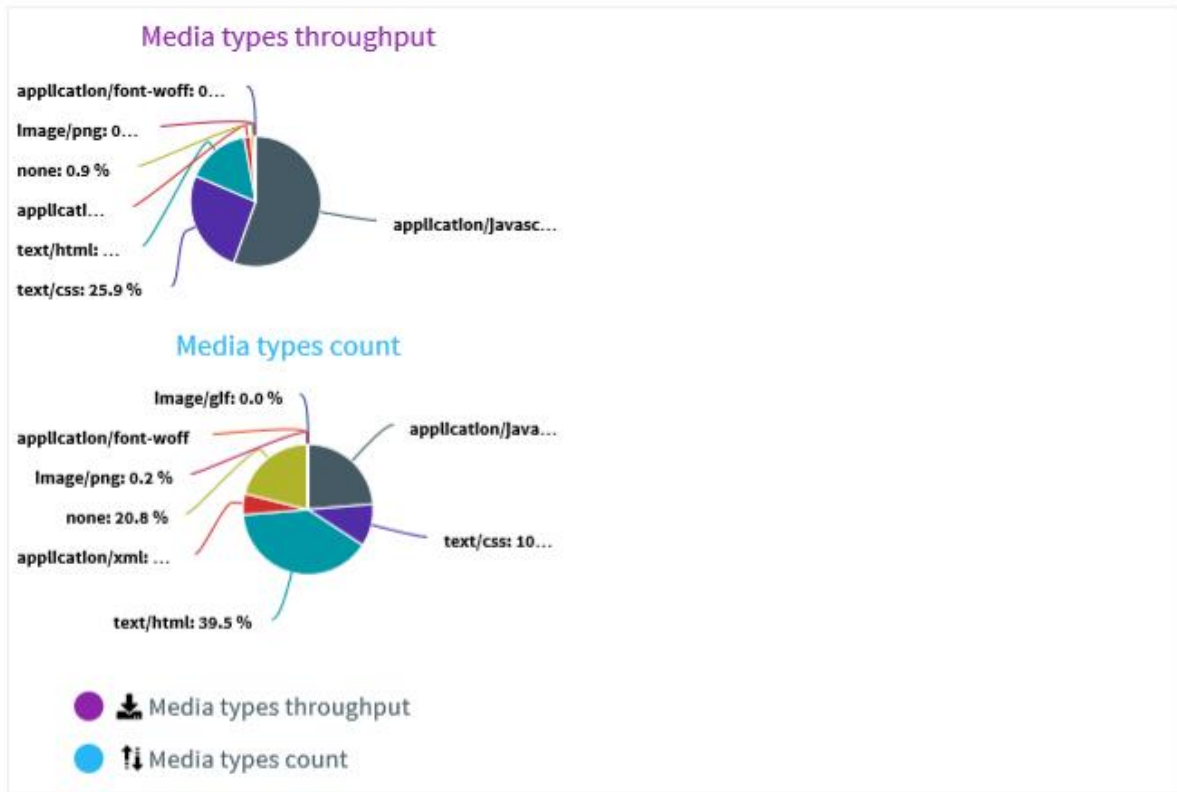


Action	Results			
	Avg. response time (sec) ▼	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
 /index.php	0.963	1.611	82999	9
 /sql_exercicio.php?i...	0.927	1.428	82603	6
 /sql_tela_esquerda...	0.887	1.425	82551	7
 /sql_janelas.php	0.866	1.392	82689	8
 /sqlscripter/ajax2.php	0.856	1.353	82140	3
 /sql_logoff.php	0.848	1.339	82071	3
 /startbootstrap/ven...	0.228	0.354	82973	1
 /sqlscripter/locale/e...	0.228	0.385	82204	2
 /sqlscripter/js/boots...	0.2	0.346	82467	0
 /sql_resultado.php	0.177	0.272	82131	1
 /sqlscripter/ajax.php	0.172	0.218	82188	1
 /sqlscripter/sql_desi...	0.168	0.287	82535	3
 /sql_tela_esquerda...	0.16	0.282	82592	0
 /sql_avaliacao.php	0.158	0.264	82674	2
 /sqlscripter/images/...	0.138	0.229	82198	1
 /sql_estatistica.php...	0.135	0.219	82122	0
 /sqlscripter/js/globa...	0.131	0.236	82491	0
 /sqlscripter/src/js/p...	0.127	0.218	82481	0
 /sqlscripter/styles/b...	0.123	0.15	82522	0

Total Items: 38

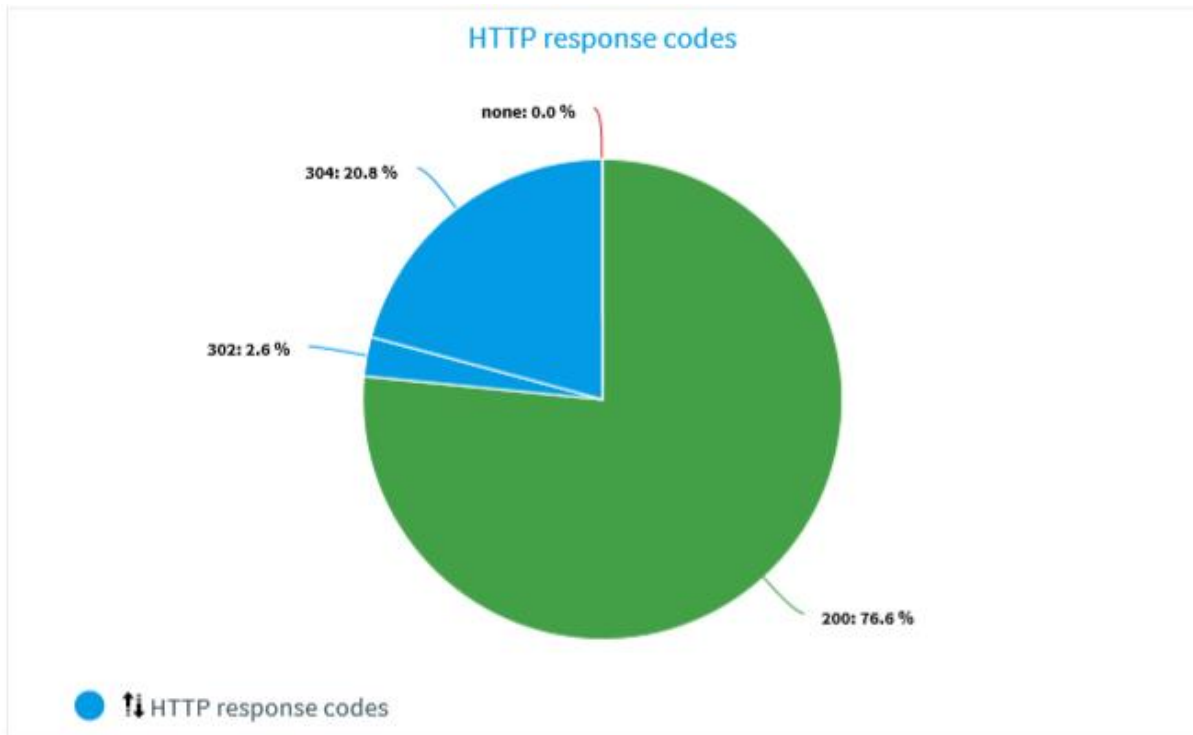
-  Avg. response time
-  Response time percentile 90
-  Hits count
-  Errors count





















3.2. Throughput



4. Error Details

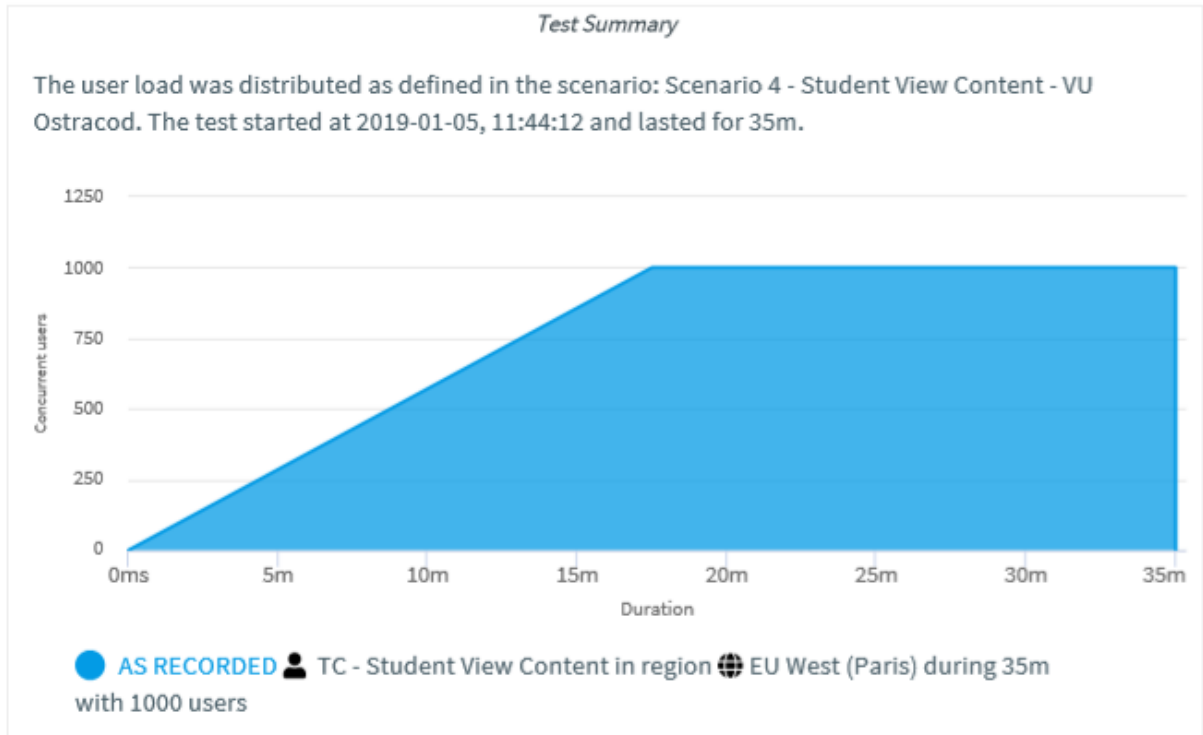
This chapter presents the error rate and detailed errors.



Errors		
Time ▲	Action	Error
	<input type="text"/>	<input type="text"/>
2019-01-05, 12:52:34	 /index.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:53:16	 /sqlscripter/locale/en.xml	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:54:21	 /sql_avalicao.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:56:08	 /sqlscripter/ajax.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:56:24	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:56:24	 /sql_logoff.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:56:24	 /startbootstrap/vendor/bootstrap/css/bootstrap.min.css	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:57:35	 /sql_tela_esquerda.php?cdProb=219&menuprincipal=...	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:57:35	 /sqlscripter/images/diamond_upholstery.png	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:57:40	 /sql_tela_esquerda.php?cdProb=219&menuprincipal=...	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:57:40	 /sqlscripter/sql_designer_erd.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:57:40	 /sql_exercicio.php?idaval=\${idaval}&conteudo=80&situ...	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:57:40	 /sql_exercicio.php?idaval=\${idaval}&conteudo=80&situ...	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:57:40	 /sql_tela_esquerda.php?cdProb=219&menuprincipal=...	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:58:31	 /sql_exercicio.php?idaval=\${idaval}&conteudo=80&situ...	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:58:32	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:58:50	 /sqlscripter/locale/en.xml	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:58:50	 /sql_tela_esquerda.php?cdProb=219&menuprincipal=...	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:58:51	 /sqlscripter/js/wwwsqldesigner.js	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:59:20	 /sql_exercicio.php?idaval=\${idaval}&conteudo=80&situ...	HTTP Status code: Non HTTP response code:...
Total Items: 56		

TC Student View Content

1. Test Summary

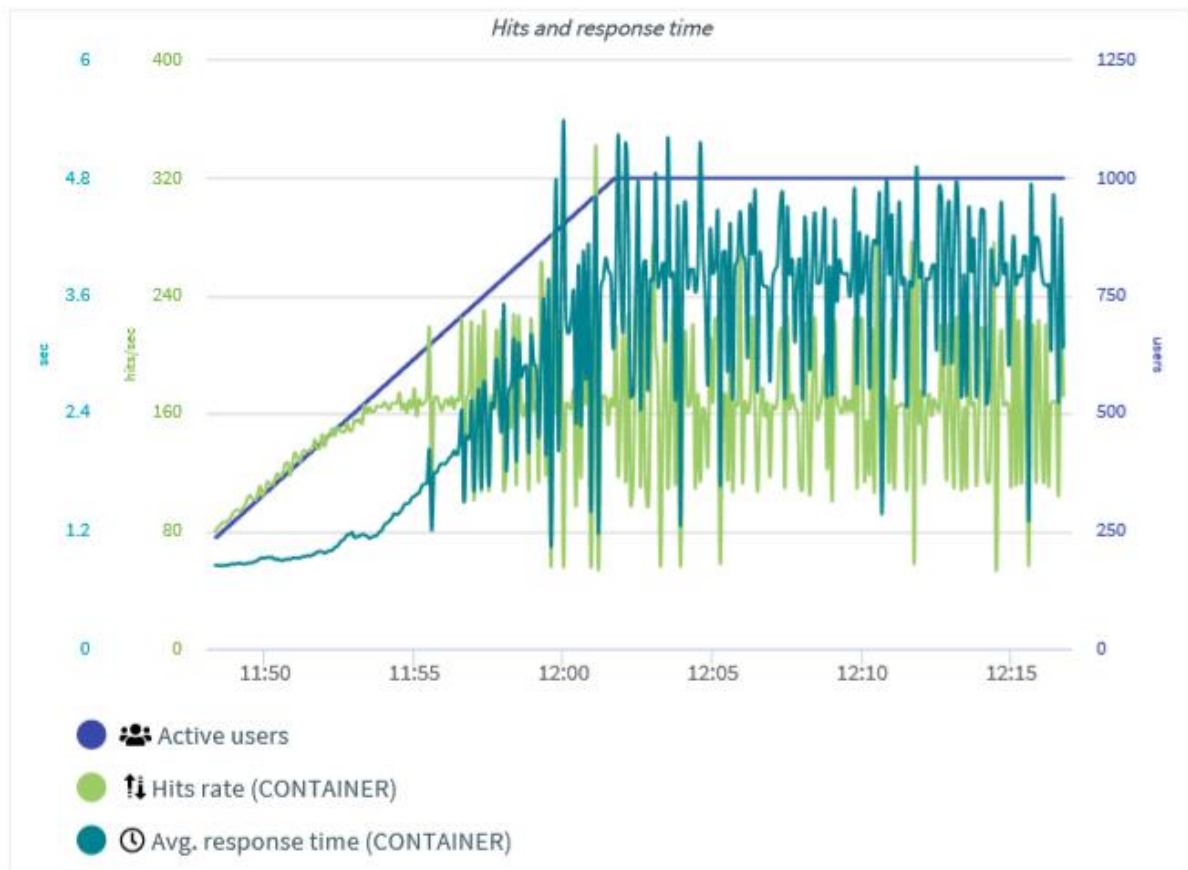


Statistics summary

	Hits count 2434731 hits
	Errors count 44 errors
	Avg. response time (CONTAINER) 2.935 sec
	Latency standard deviation (CONTAINER) 1.541 sec
	Received Bytes rate 1.5 MB/sec
	Apdex on response time (CONTAINER) 0.589 apdex

2. Results Summary

2.1. Hit rate and response time under load














2.2. Request details



<i>Results tree</i>					
	Action	Avg. response time (sec)	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
	 AS RECORDED - T...				
	 TC - Student L...	2.927	4.083	90705	17
	 TC - View Cont...	4.881	6.661	90019	22
	 TC - Logout	0.996	1.411	89813	5

Total Items: 31 (Showing Items: 4)

-  ⌚ Avg. response time
-  ⌚ Response time percentile 90
-  ↑↓ Hits count
-  ! Errors count

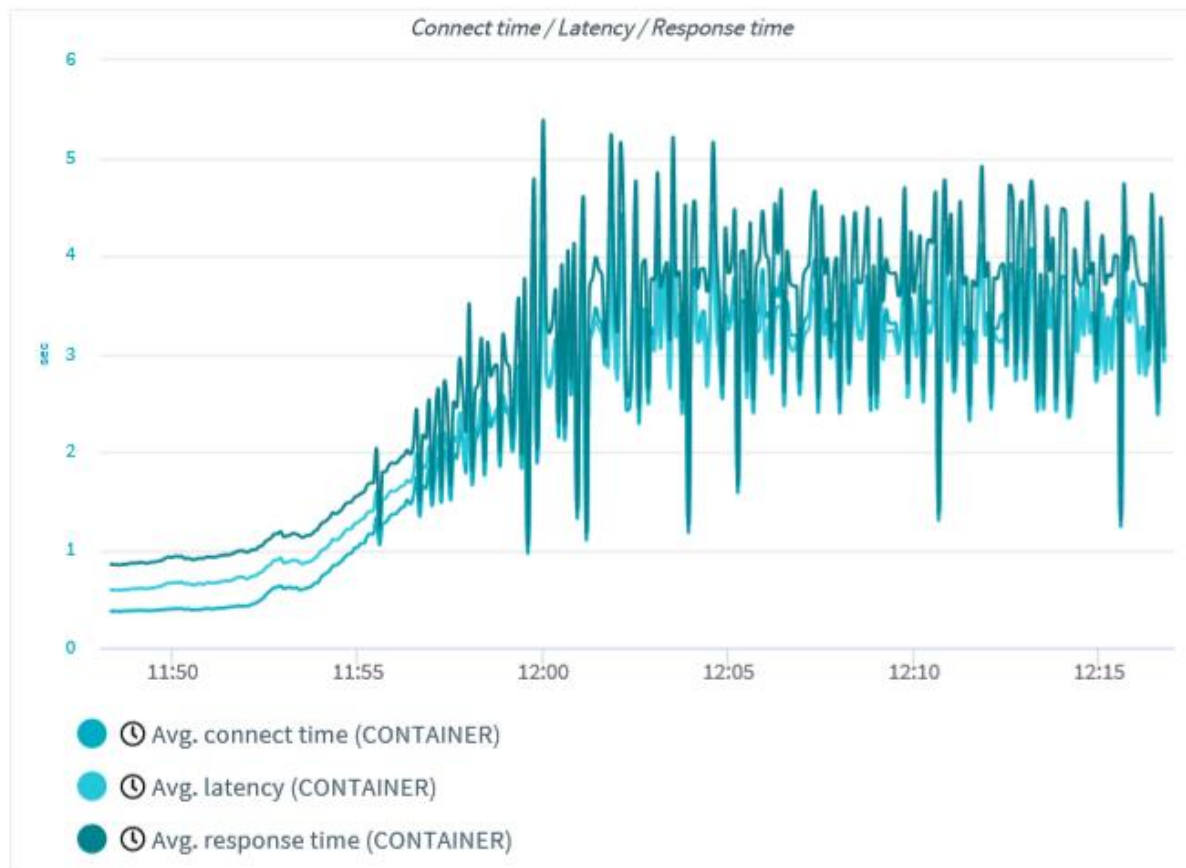
Action	Results			
	Avg. response time (sec) ▼	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
 /sql_main.php?men...	1.045	1.47	90216	5
 /sql_avaliacao.php	1.005	1.438	90261	2
 /sql_logoff.php	0.996	1.411	89813	5
 /sql_main.php	0.99	1.41	90205	2
 /sql_janelas.php	0.986	1.432	90638	8
 /conteudo/itemdina...	0.983	1.43	89878	6
 /index.php	0.929	1.325	90559	7
 /imagens/fundoBarr...	0.188	0.311	90148	2
 /imagens/ftv2plastn...	0.174	0.285	89948	1
 /imagens/topo.jpg	0.172	0.289	90160	3
 /ua.js	0.146	0.241	90194	0
 /conteudo/itemdina...	0.139	0.216	90168	1
 /imagens/ftv2vertin...	0.091	0.135	90172	0
 /imagens/anterior.jpg	0.086	0.129	90191	0
 /imagens/ftv2node.gif	0.081	0.118	90177	0
 /imagens/ftv2mnod...	0.081	0.117	90177	1
 /imagens/ftv2pnode...	0.079	0.116	89945	0
 /ftiens4.js	0.077	0.113	90192	0
 /imagens/ftv2doc.gif	0.072	0.101	90187	0

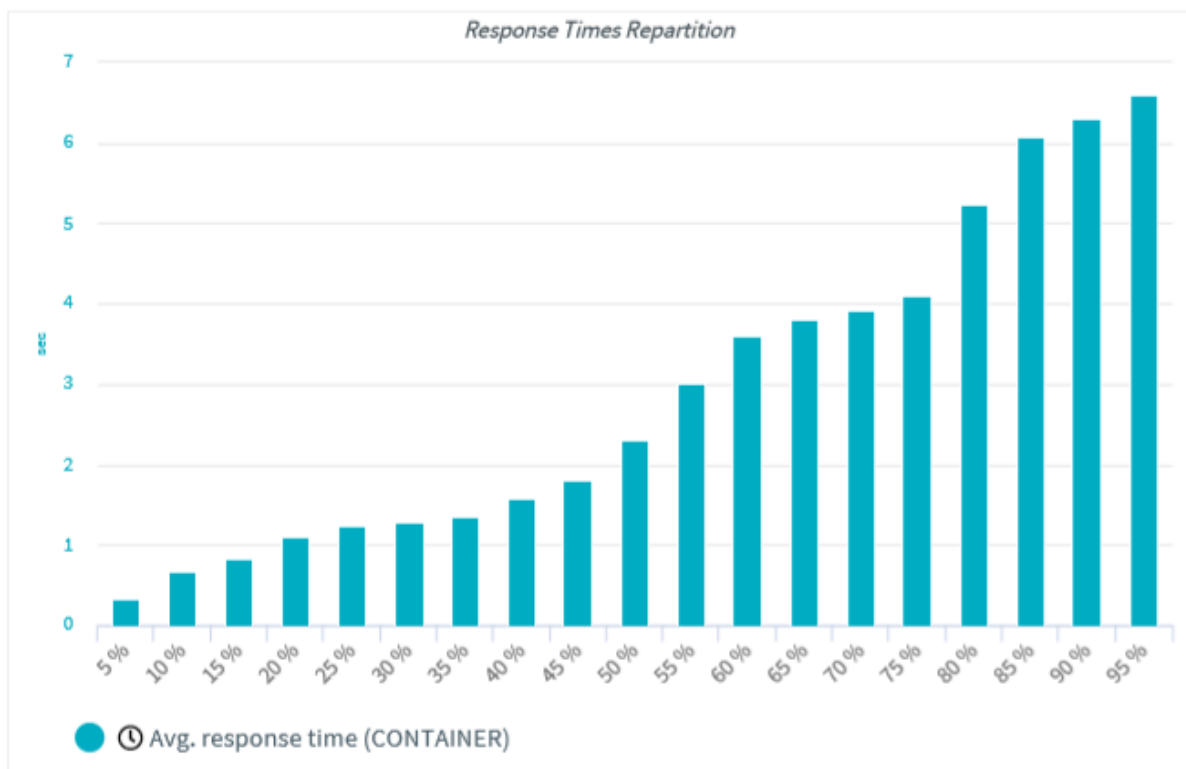
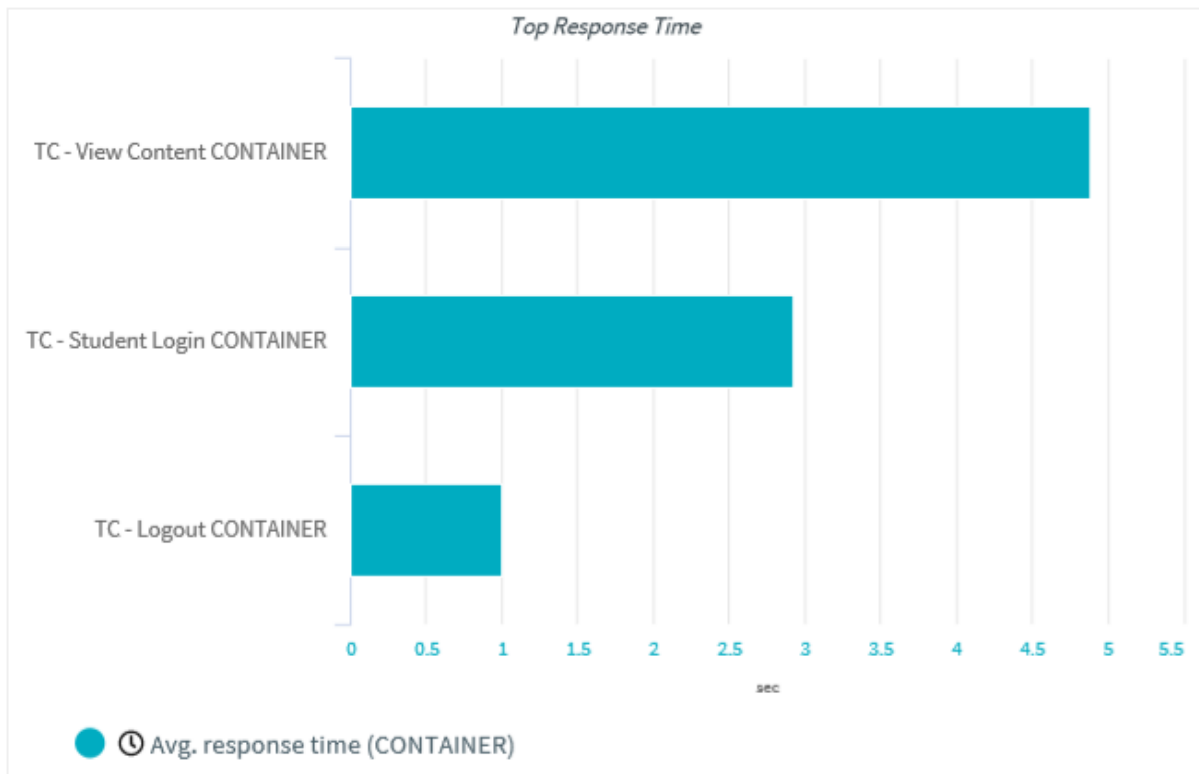
Total Items: 27

-  ⌚ Avg. response time
-  ⌚ Response time percentile 90
-  ↕ Hits count
-  ! Errors count

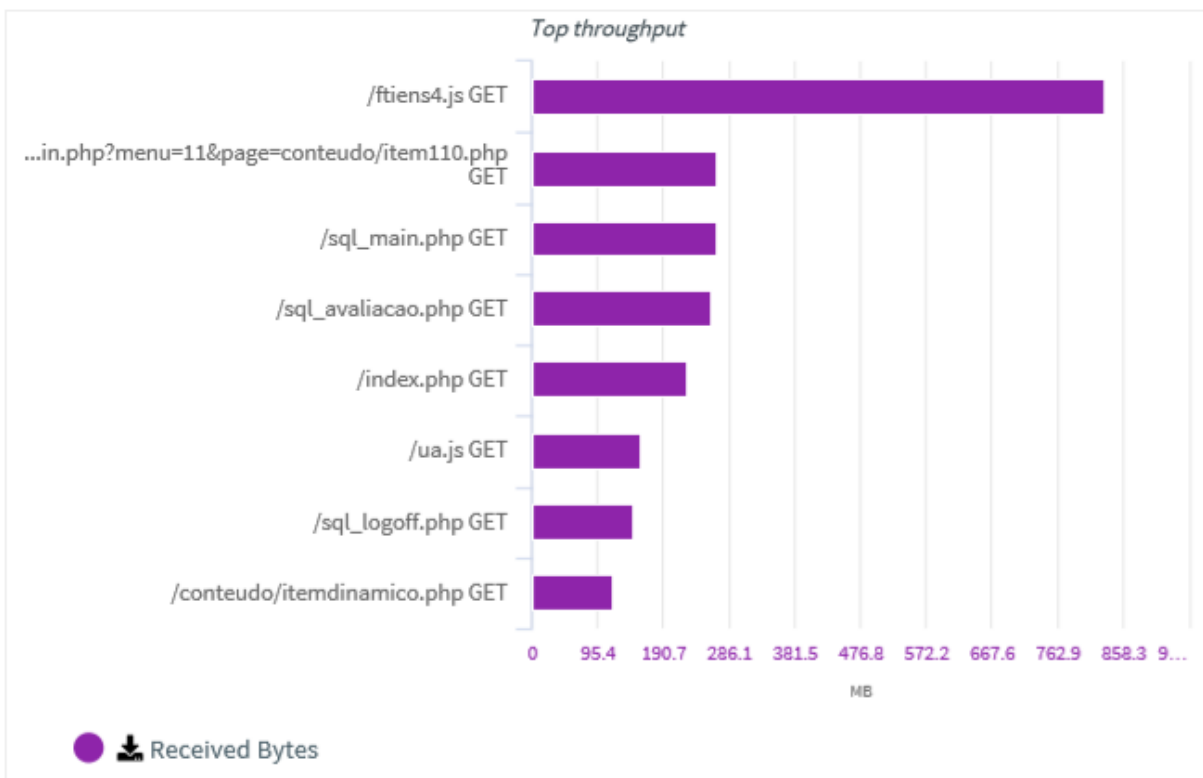
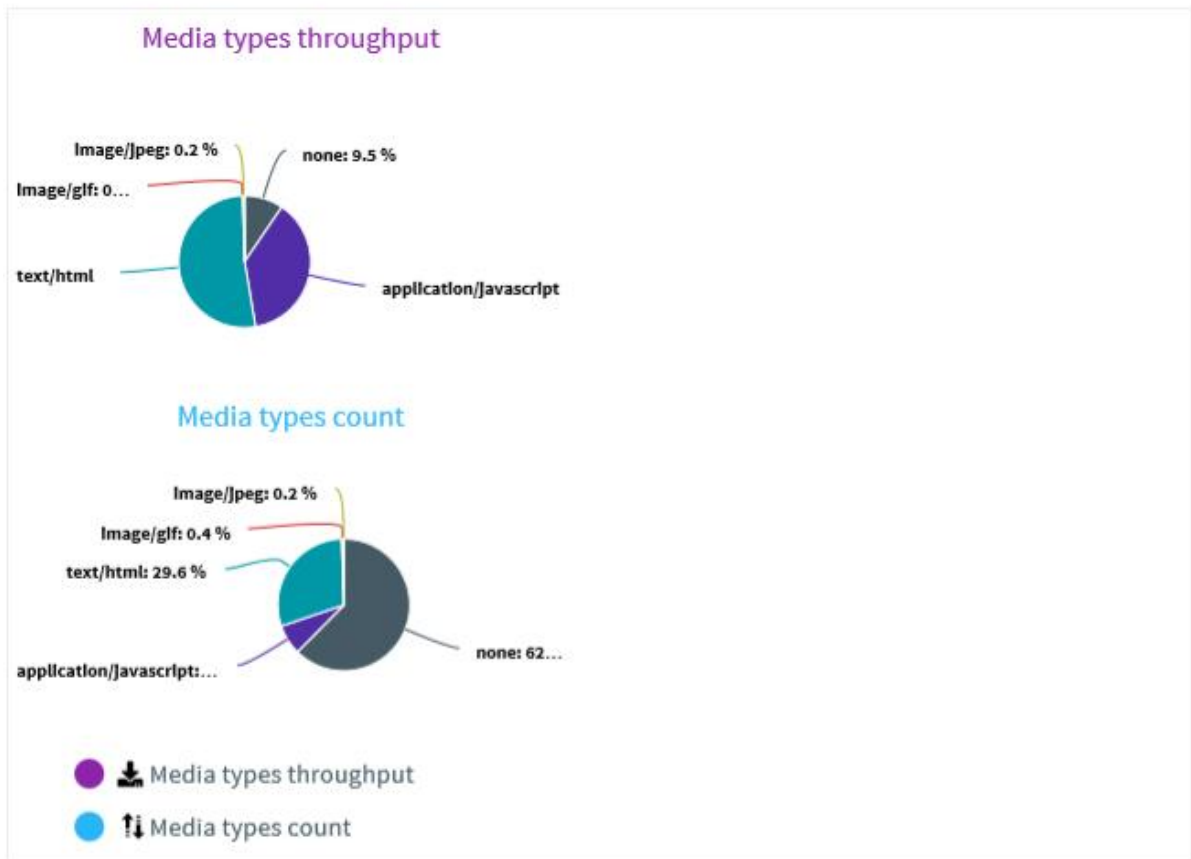
3. Result breakdown

3.1. Response time and percentiles



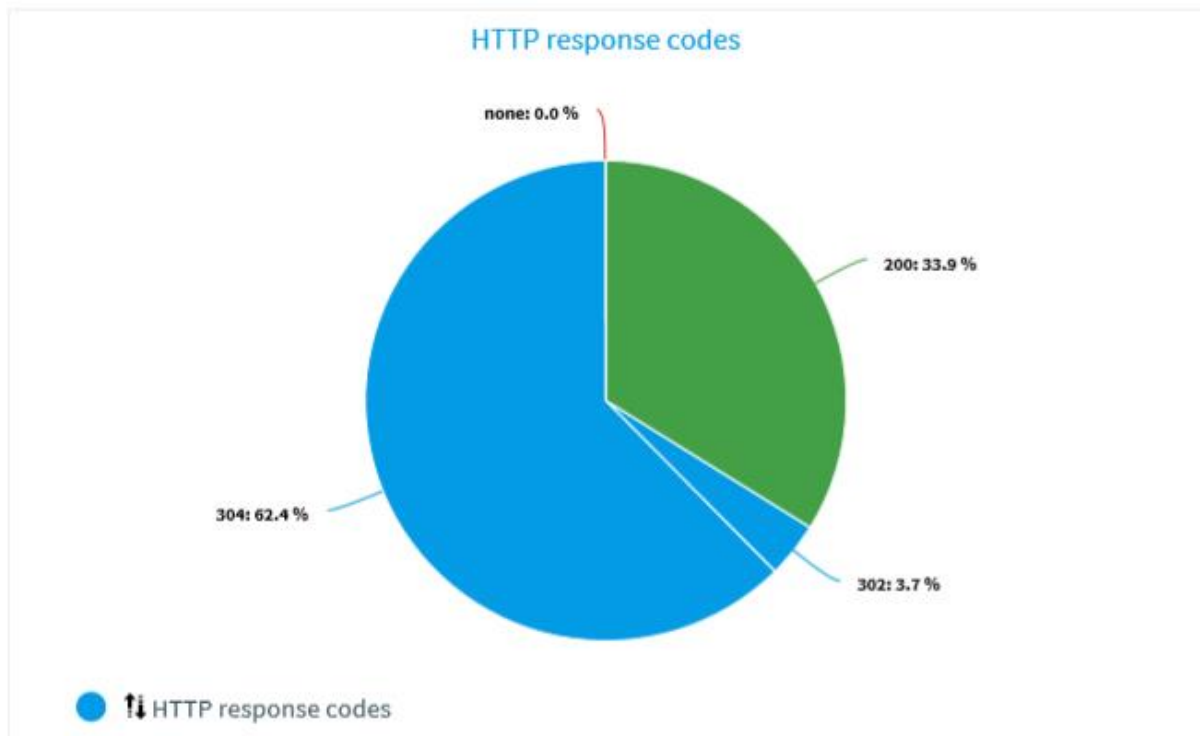
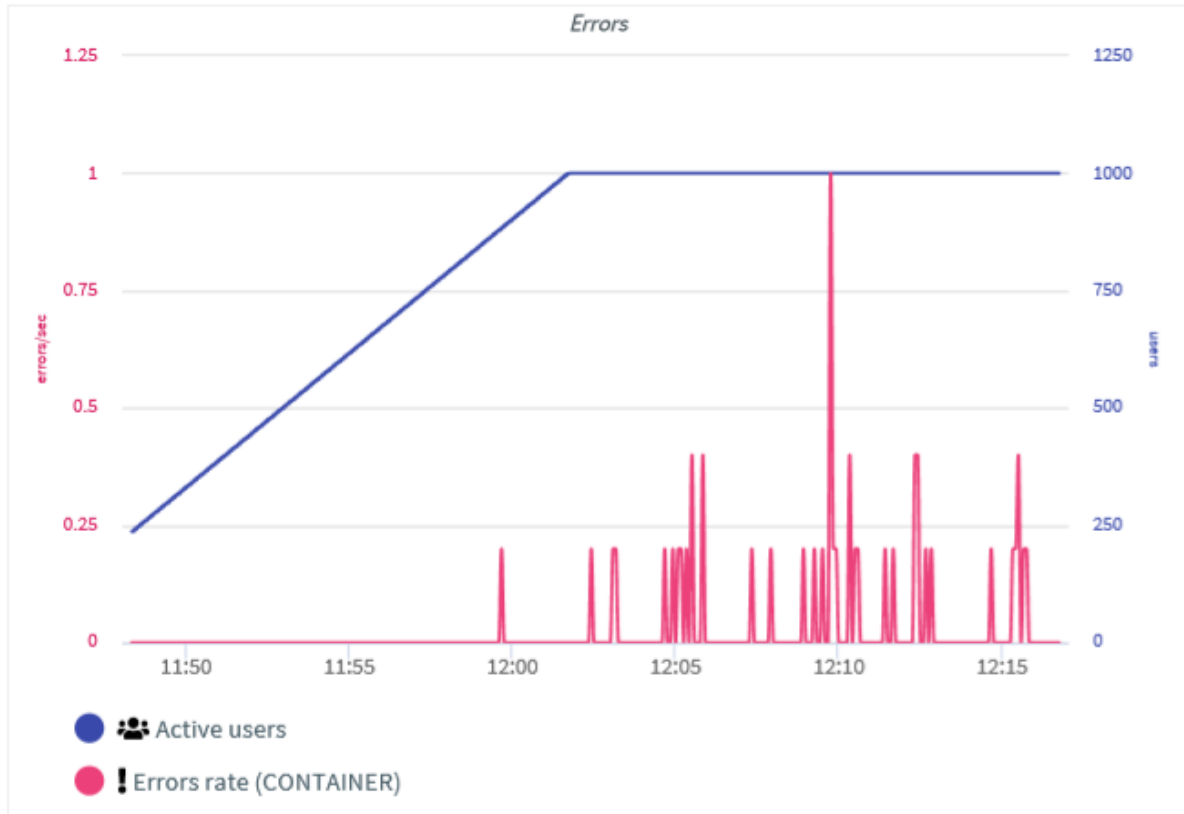






















3.2. Throughput



4. Error Details

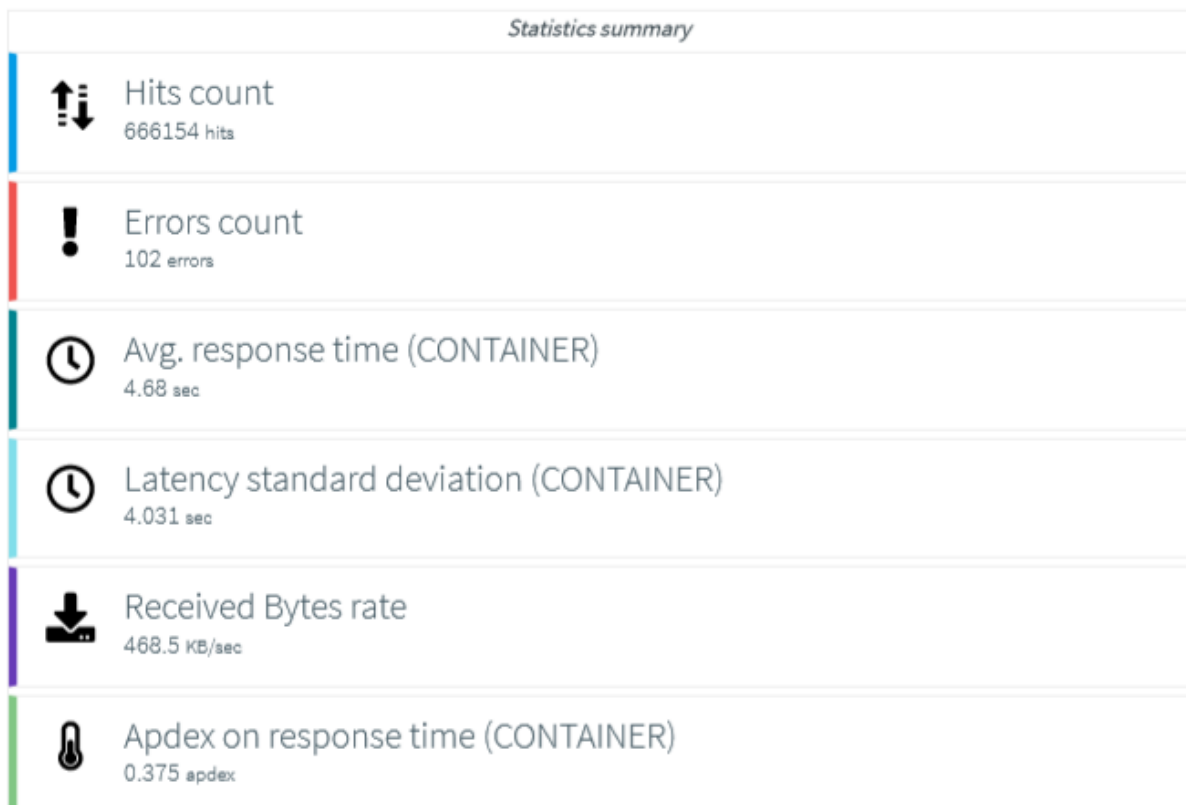
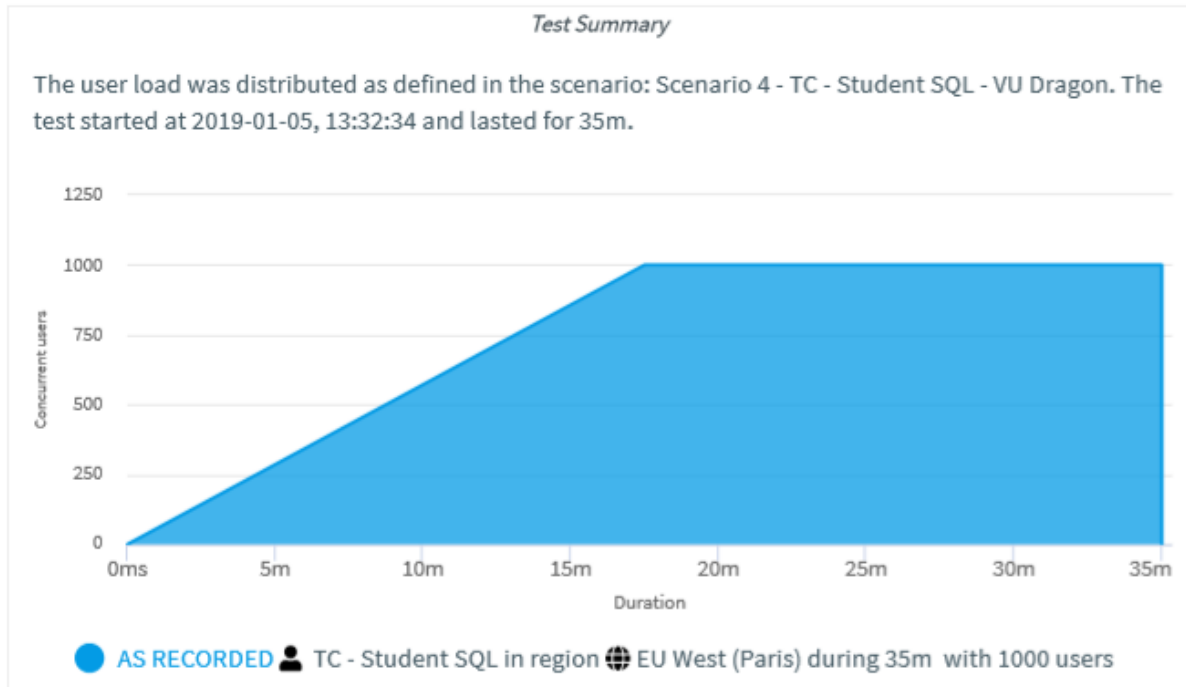
This chapter presents the error rate and detailed errors.



Errors		
Time ▲	Action	Error
	<input type="text"/>	<input type="text"/>
2019-01-05, 11:58:45	 /sql_logoff.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:01:23	 /imagens/fundoBarraPrincipal.jpg	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:02:14	 /conteudo/itemdinamico.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:02:19	 /imagens/fundoBarraPrincipal.jpg	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:03:42	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:03:58	 /sql_main.php?menu=11&page=conteudo/item110.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:04:01	 /index.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:04:09	 /sql_logoff.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:04:15	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:04:32	 /conteudo/itemdinamico.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:04:33	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:04:50	 /index.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:04:50	 /index.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:06:23	 /conteudo/itemdinamico.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:07:01	 /conteudo/itemdinamico.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:07:55	 /index.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:08:17	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:08:34	 /imagens/topo.jpg	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:08:44	 /sql_logoff.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 12:08:48	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
Total Items: 52		

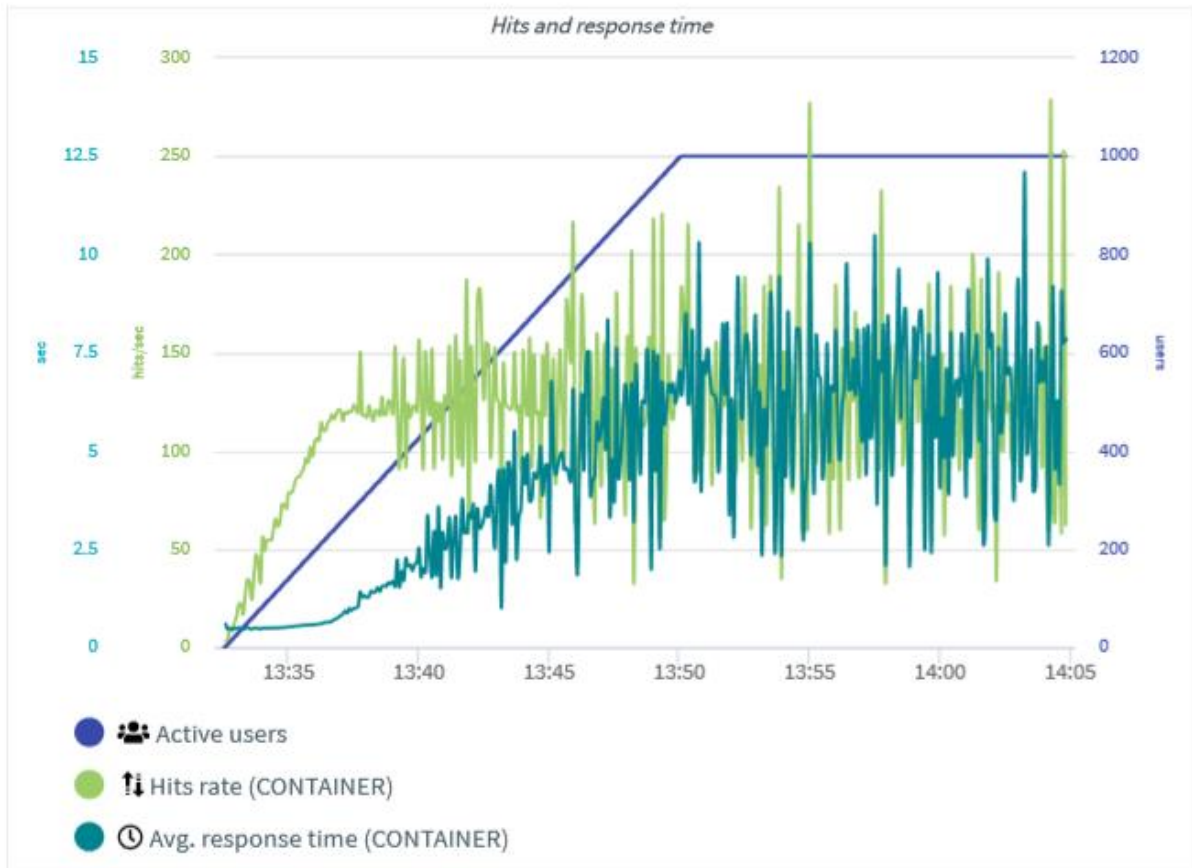
TC – Student SQL

1. Test Summary














2. Results Summary









2.1. Hit rate and response time under load











2.2. Request details





<i>Results tree</i>					
	Action	Avg. response time (sec)	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
	 AS RECORDED - T...				
	 TC - Student L...	2.083	3.368	56044	31
	 TC - Student S...	9.9	14.951	55889	37
	 TC - Student S...	5.339	7.704	55194	24
	 TC - Student L...	1.356	2.257	54951	10

Total Items: 17 (Showing Items: 5)

-   Avg. response time
-   Response time percentile 90
-   Hits count
-   Errors count

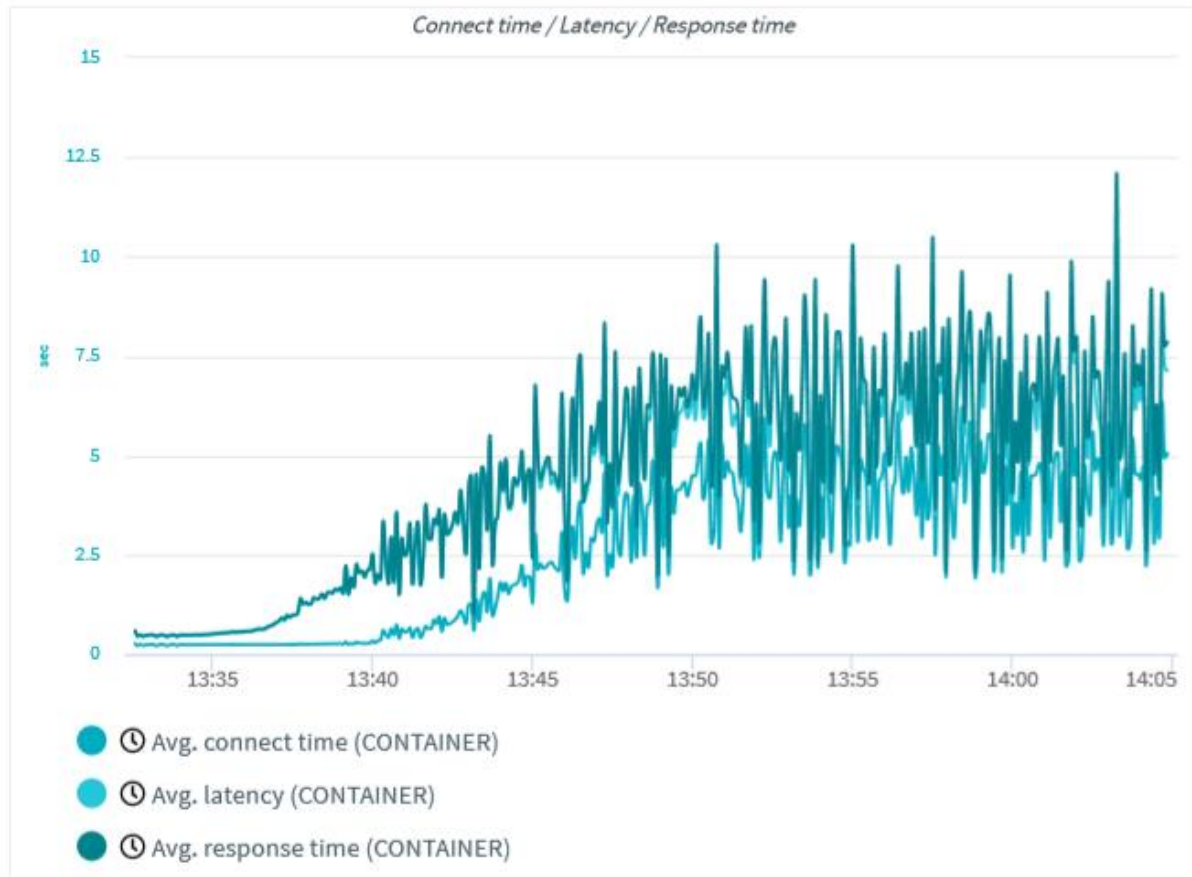
Action	Results			
	Avg. response time (sec) ▼	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
 /sql_resultado.php	3.922	5.397	55249	8
 /sql_resultado.php	2.793	3.712	55491	1
 /sql_exercicio.php?i...	2.139	3.265	55749	9
 /sql_tela_esquerda....	1.734	2.752	55618	4
 /sql_janelas.php	1.504	2.346	56062	7
 /sql_estatistica.php	1.466	2.511	55446	11
 /sql_estatistica.php?...	1.419	2.413	55024	16
 /sql_logoff.php	1.356	2.257	54951	10
 /sql_tela_esquerda....	0.781	1.465	55701	8
 /sql_avalicao.php	0.582	1.066	55863	24
 /imagens/Analisar.jpg	0.576	1.092	55589	3
 /imagens/barHistori...	0.416	0.839	55411	1

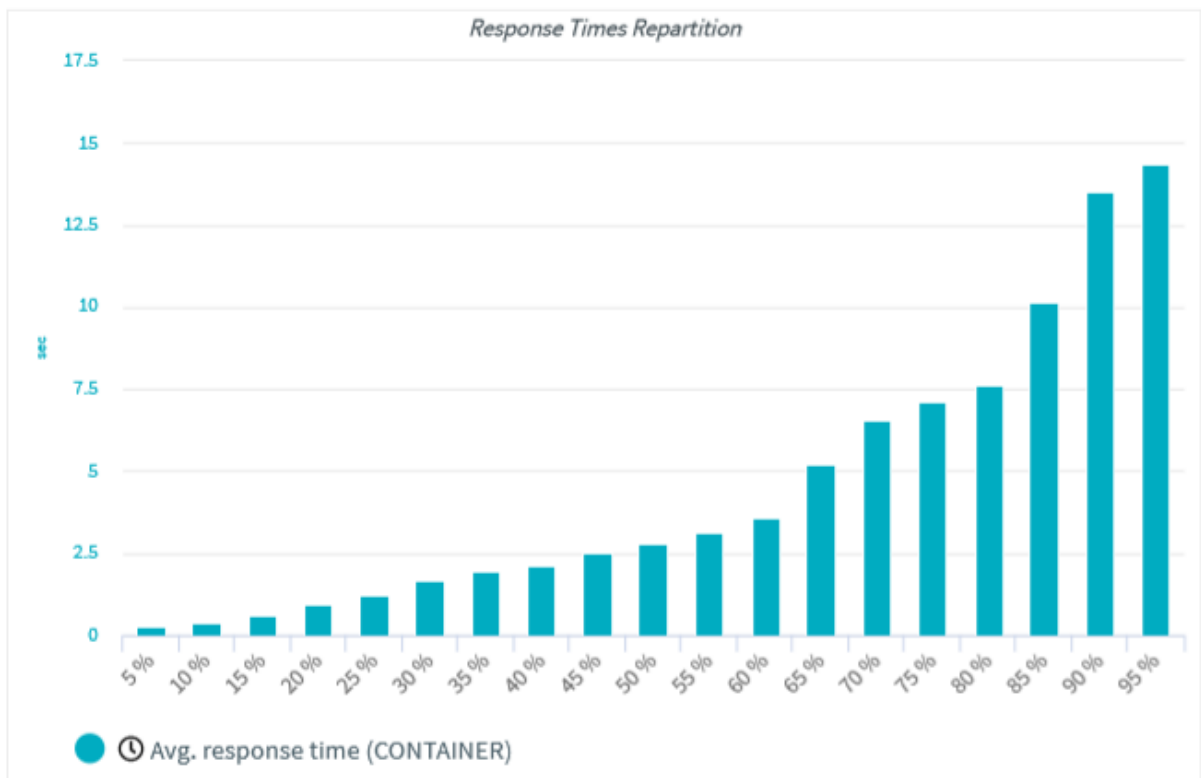
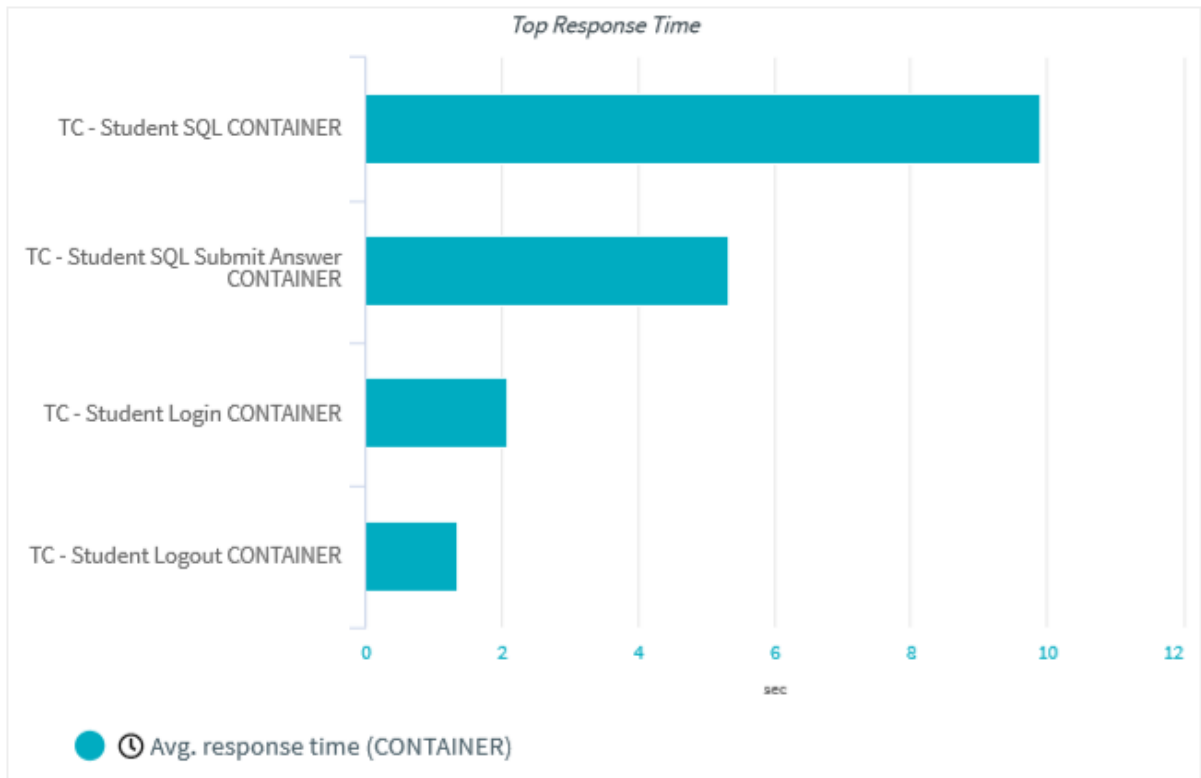
Total Items: 12

-  ⌚ Avg. response time
-  ⌚ Response time percentile 90
-  📈 Hits count
-  ! Errors count

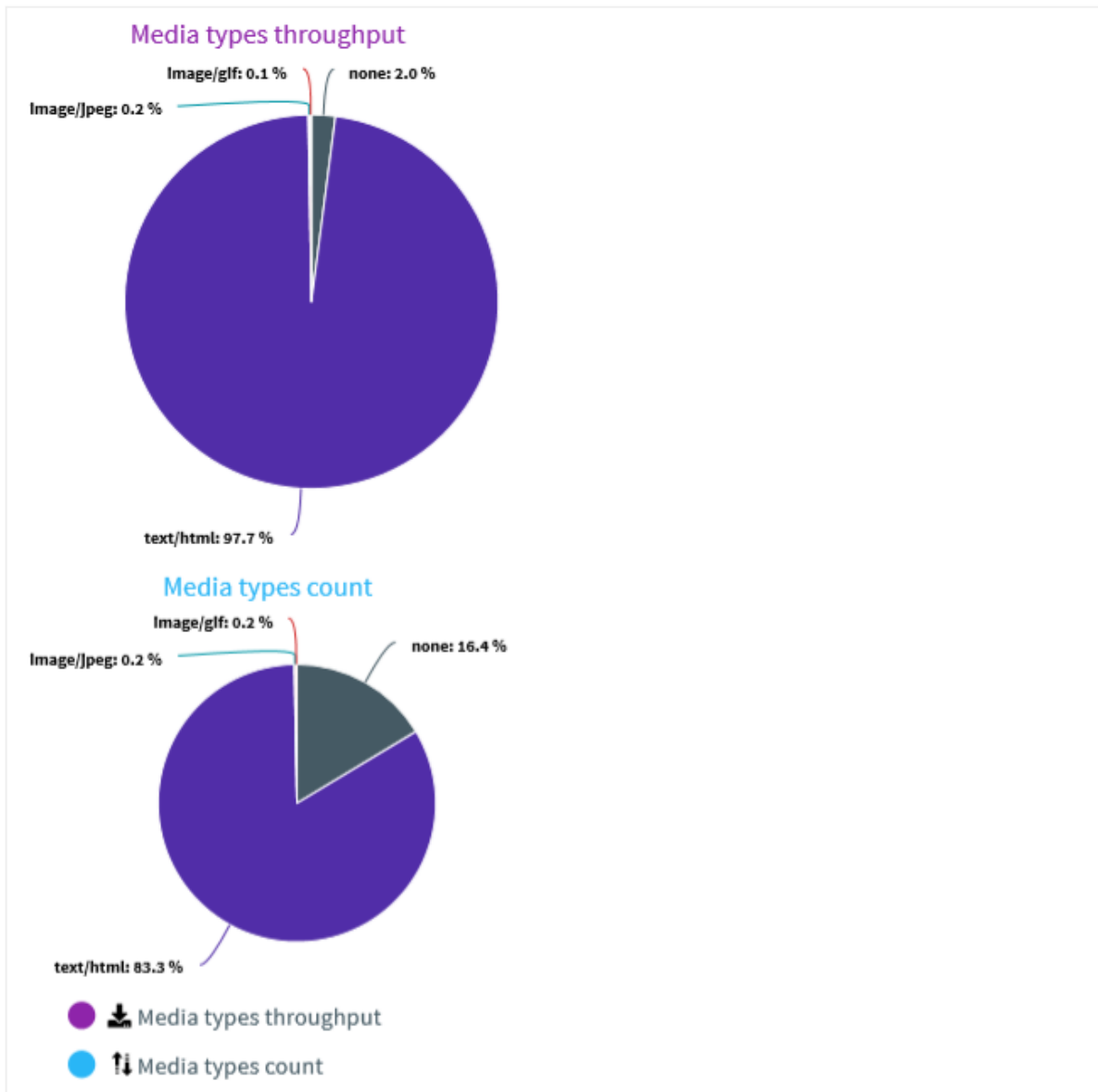
3. Result breakdown

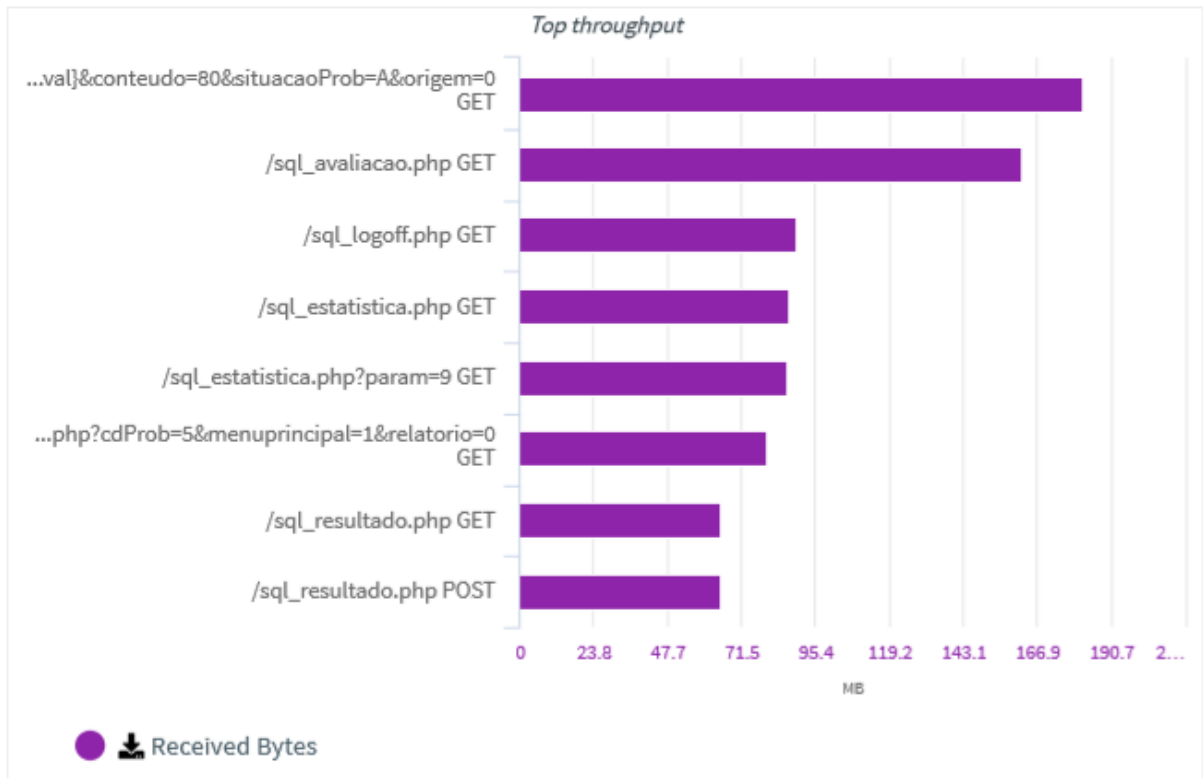
3.1. Response time and percentiles





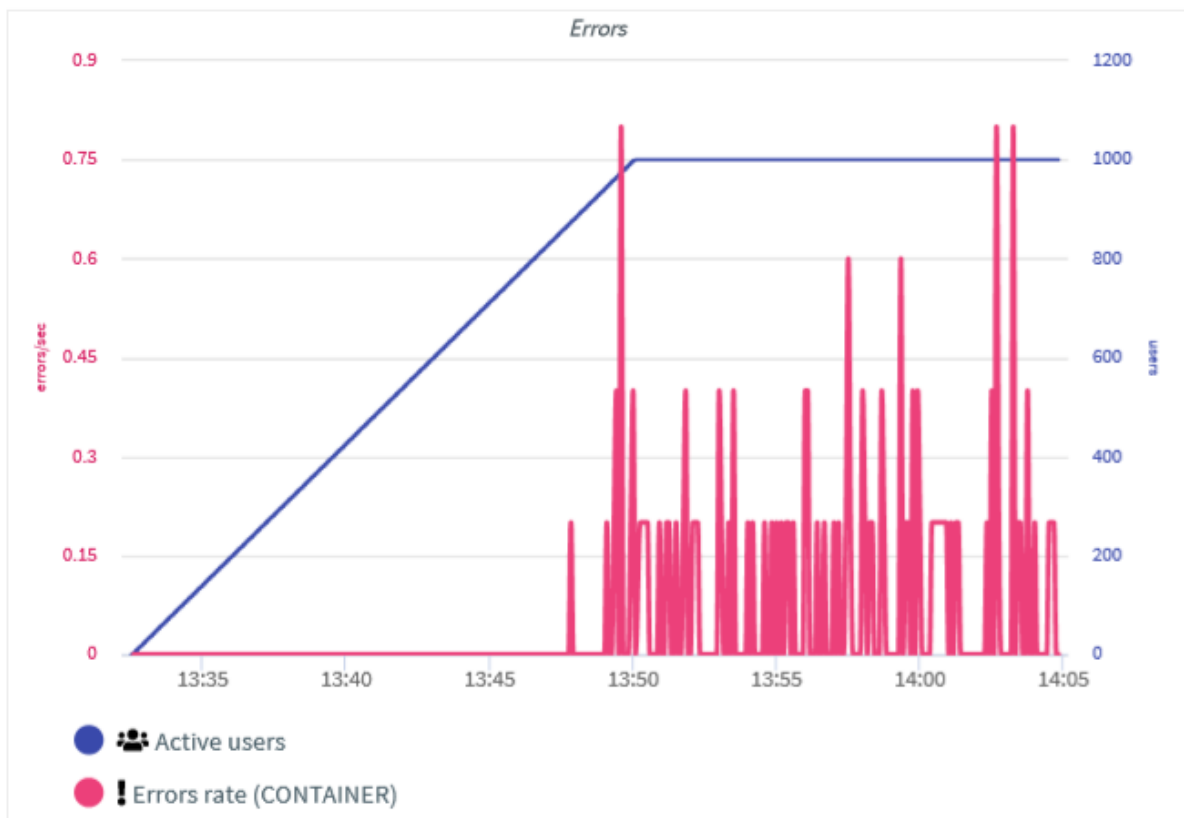
3.2. Throughput

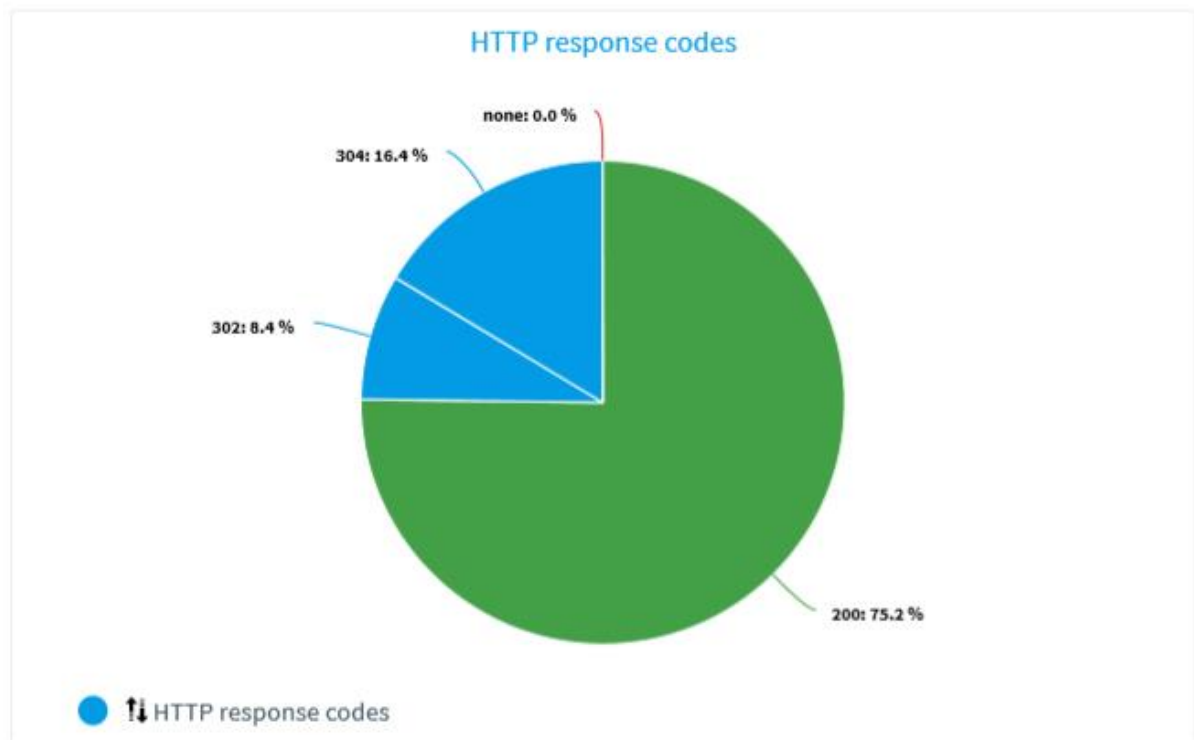
























4. Error Details

This chapter presents the error rate and detailed errors.

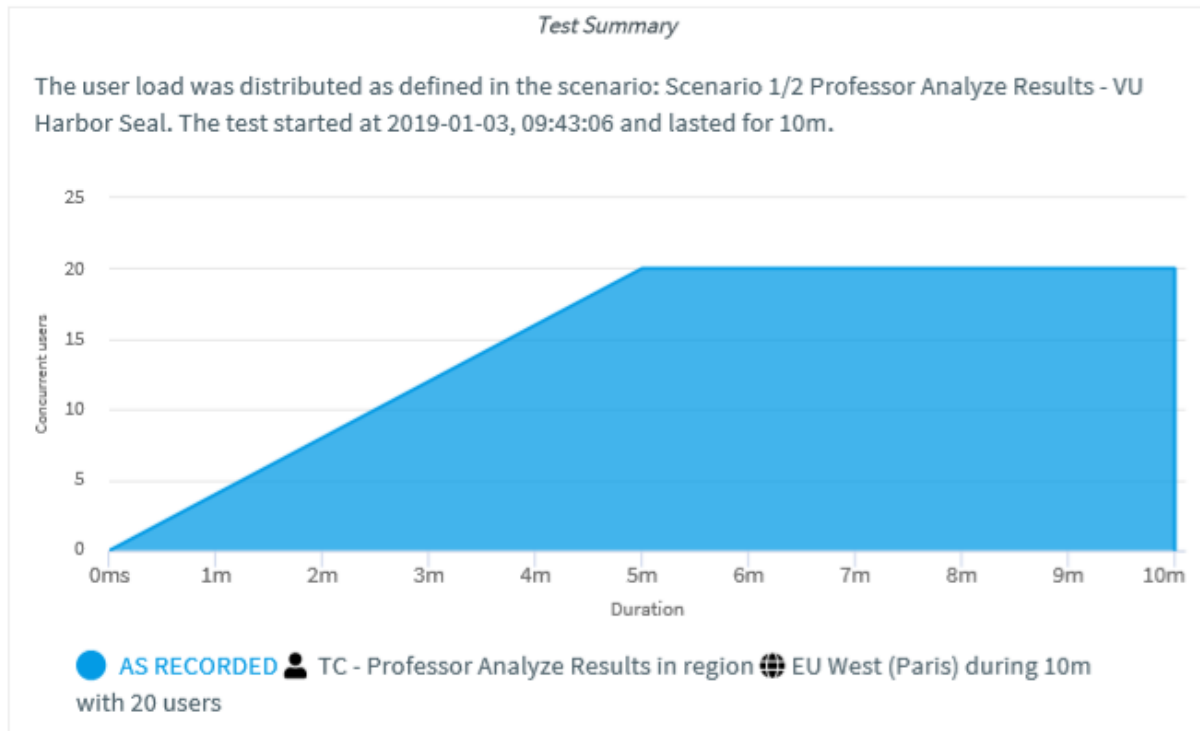










Errors		
Time ▲	Action	Error
	<input type="text"/>	<input type="text"/>
2019-01-05, 14:00:28	 /imagens/Analisar.jpg	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:00:47	 /sql_estatistica.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:20	 /sql_avaliacao.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:23	 /sql_tela_esquerda.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:38	 /sql_resultado.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:38	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:40	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:41	 /sql_avaliacao.php	HTTP Status code: 302 Found
2019-01-05, 14:01:44	 /sql_resultado.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:44	 /sql_tela_esquerda.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:01:44	 /sql_avaliacao.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:02:24	 /sql_avaliacao.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:02:24	 /sql_estatistica.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:02:31	 /sql_exercicio.php?idaval=\${idaval}&conteudo=80&situ...	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:02:33	 /sql_exercicio.php?idaval=\${idaval}&conteudo=80&situ...	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:02:34	 /sql_estatistica.php?param=9	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:02:34	 /sql_estatistica.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:02:58	 /sql_estatistica.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:03:31	 /sql_janelas.php	HTTP Status code: Non HTTP response code:...
2019-01-05, 14:03:53	 /sql_avaliacao.php	HTTP Status code: 302 Found
Total Items: 100		

TC – Professor Analyze Results

1. Test Summary



Statistics summary

	Hits count 36825 hits
	Errors count 0 errors
	Avg. response time (CONTAINER) 0.913 sec
	Latency standard deviation (CONTAINER) 1.035 sec
	Received Bytes rate 227.9 KB/sec
	Apdex on response time (CONTAINER) 0.814 apdex

2. Results Summary

2.1. Hit rate and response time under load












2.2. Request details

<i>Results tree</i>					
	Action	Avg. response time (sec)	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
	AS RECORDED - T...				
	TC - Professor ...	0.444	0.455	425	0
	TC - Professor ...	0.32	0.326	425	0
	EV - Professor ...	0.065	0.067	423	0
	TC - Professor ...	1.189	1.211	417	0
	TC - Professor ...	0.463	0.476	417	0
	EV - Professor ...	4.343	4.418	409	0
	EV - Professor ...	0.29	0.294	406	0
	TC - Professor ...	0.23	0.236	406	0

Total Items: 98 (Showing Items: 9)

- ⌚ Avg. response time
- ⌚ Response time percentile 90
- ⬆️⬆️ Hits count
- ❗ Errors count

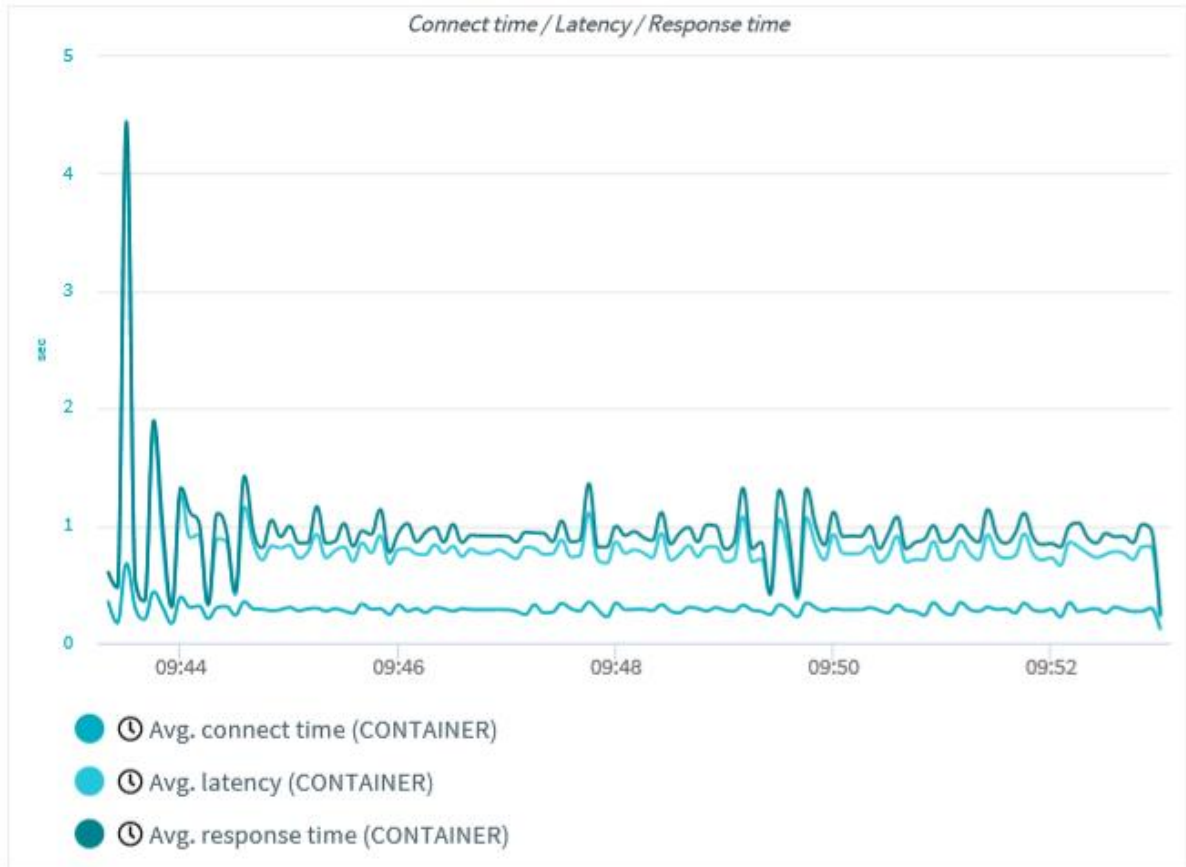
Action	Results			
	Avg. response time (sec) ▾	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
 /Rel/sql_analisa_res...	0.311	0.318	420	0
 /sql_exercicio.php?c...	0.267	0.276	417	0
 /sql_atalho.php	0.252	0.257	425	0
 /sql_tela_esquerda....	0.238	0.245	417	0
 /sql_tela_esquerda....	0.236	0.241	412	0
 /sql_resultado.php	0.233	0.238	407	0
 /sql_logoff.php	0.23	0.236	406	0
 /sql_professor.php?...	0.229	0.237	423	0
 /sql_janelas.php	0.226	0.229	425	0
 /Rel/sql_analisa_res...	0.225	0.229	420	0
 /sqlscripter/backen...	0.224	0.23	409	0
 /	0.219	0.223	425	0
 /scripts/estilosprof.c...	0.211	0.217	419	0
 /sqlscripter/locale/e...	0.17	0.215	413	0
 /sqlscripter/styles/b...	0.101	0.103	414	0
 /sqlscripter/styles/b...	0.101	0.104	412	0
 /sqlscripter/js/jquer...	0.1	0.104	413	0
 /sqlscripter/js/jquer...	0.1	0.105	412	0
 /imagens/barHistori...	0.086	0.149	413	0

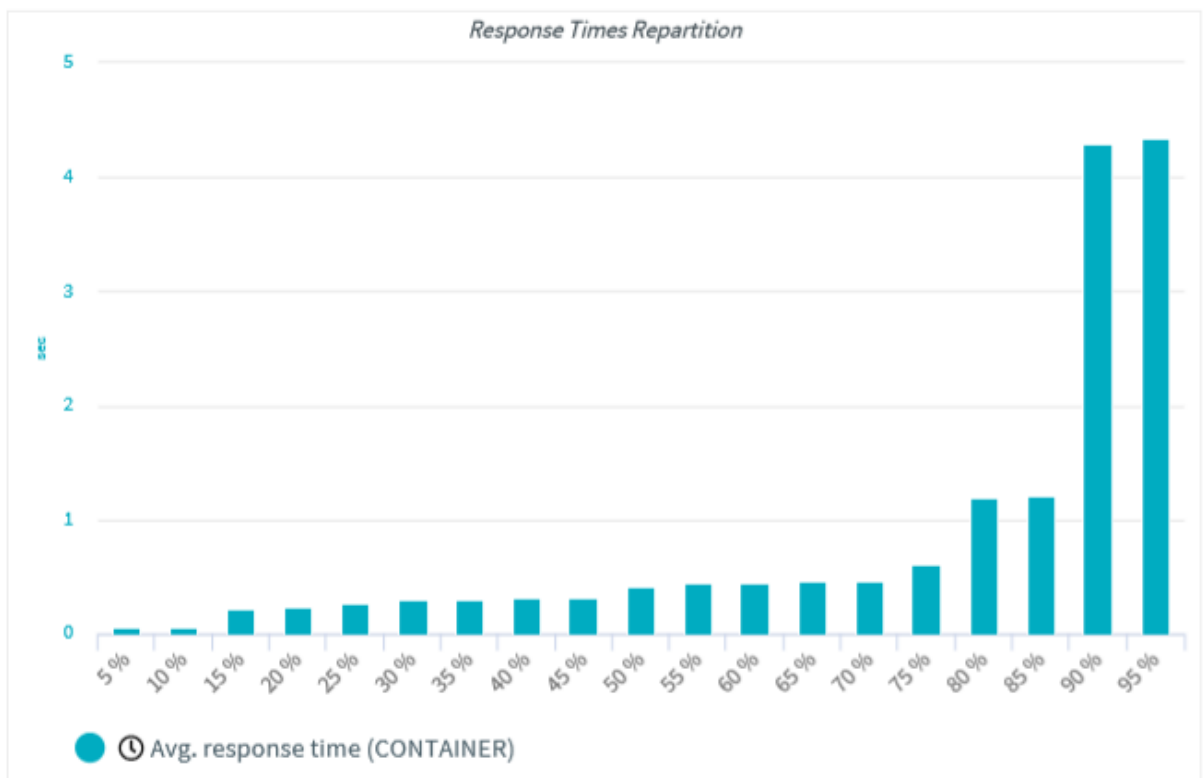
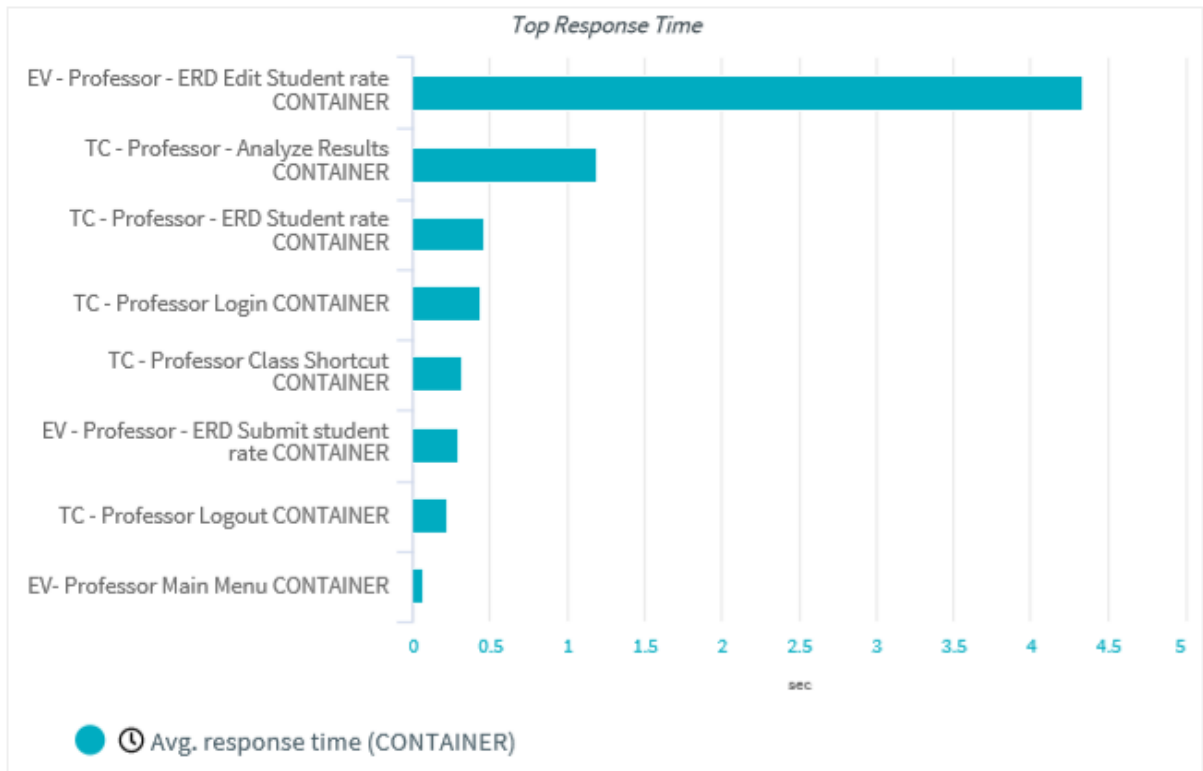
Total Items: 89

- ⌚ Avg. response time
- ⌚ Response time percentile 90
- ⬆️ Hits count
- ! Errors count

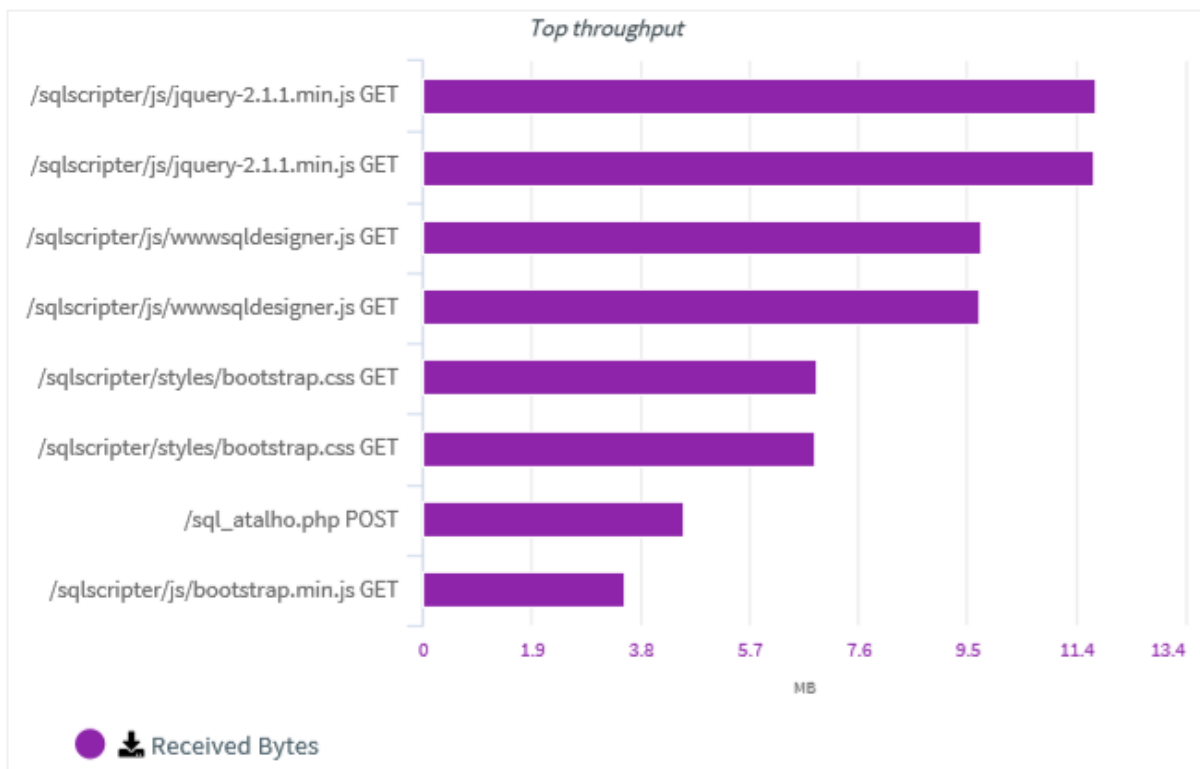
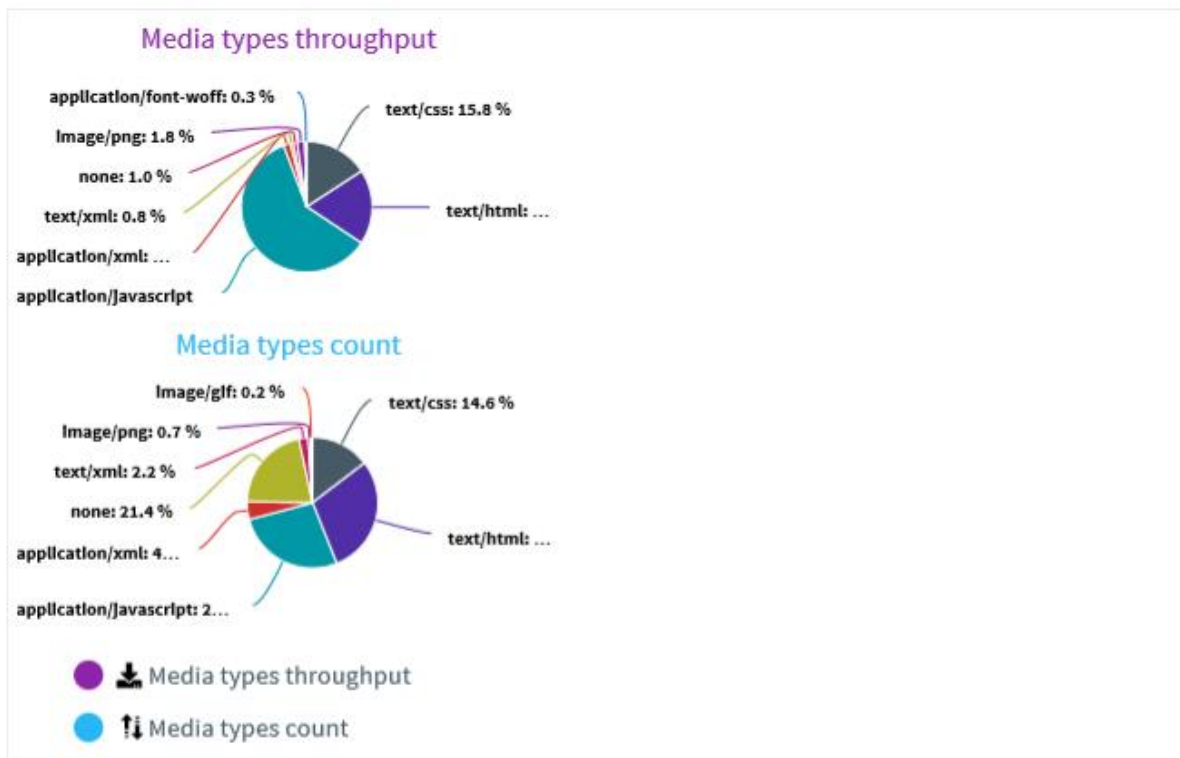
3. Result breakdown

3.1. Response time and percentiles



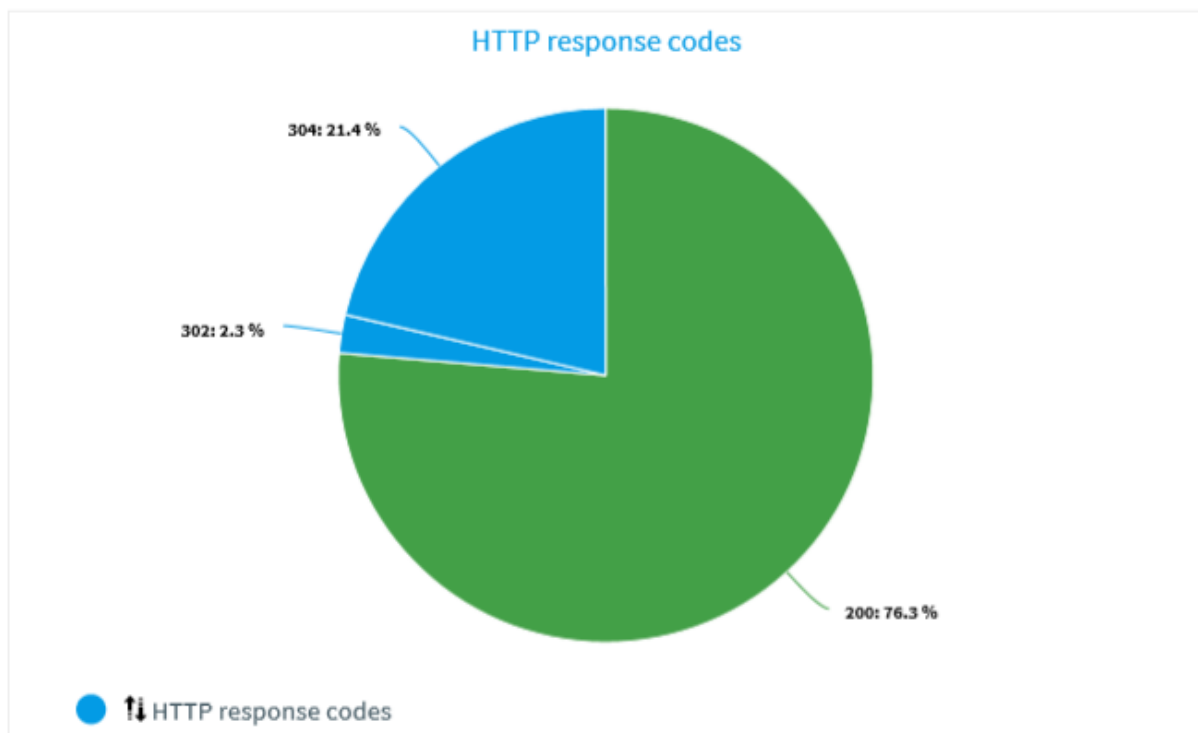
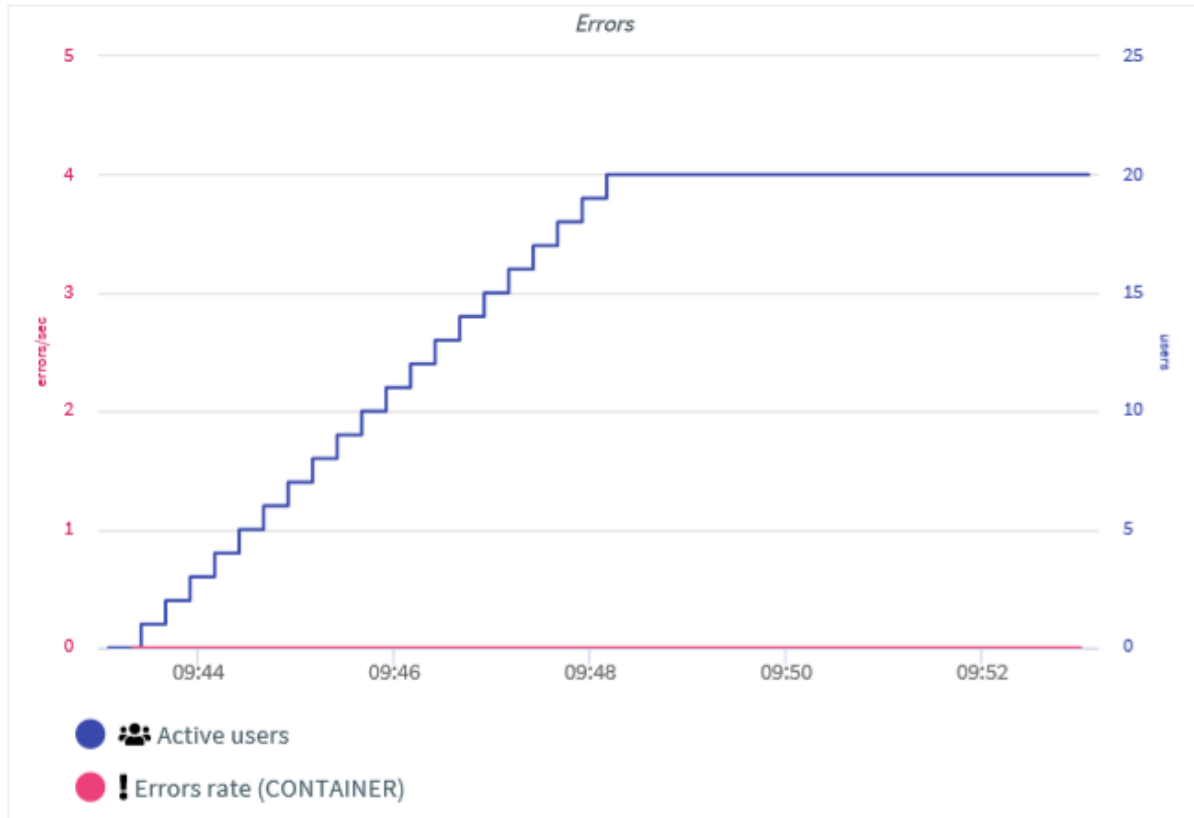


3.2. Throughput



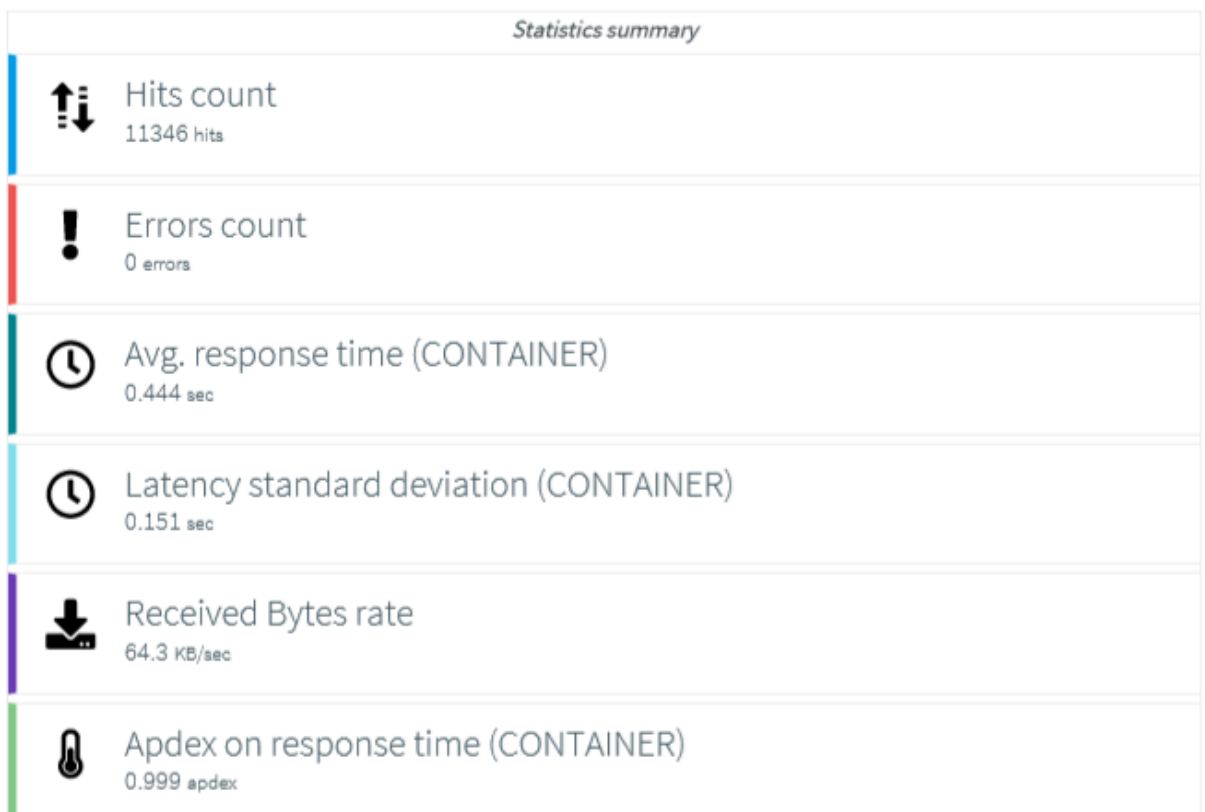
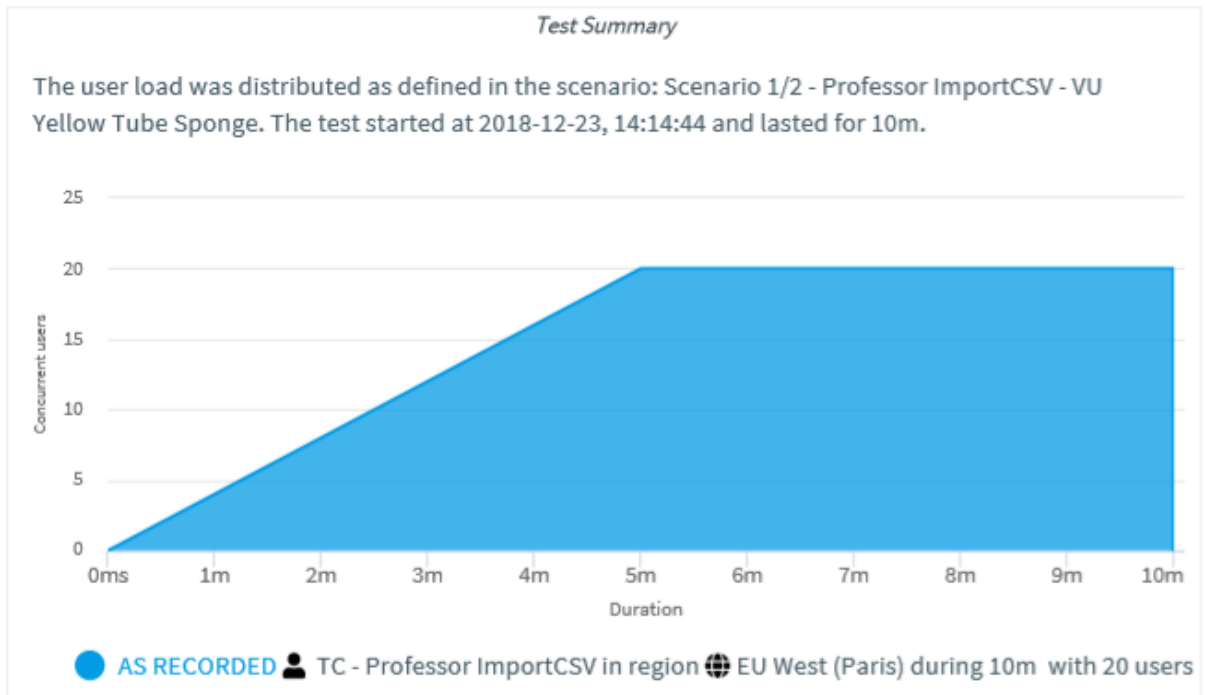
4. Error Details

This chapter presents the error rate and detailed errors.



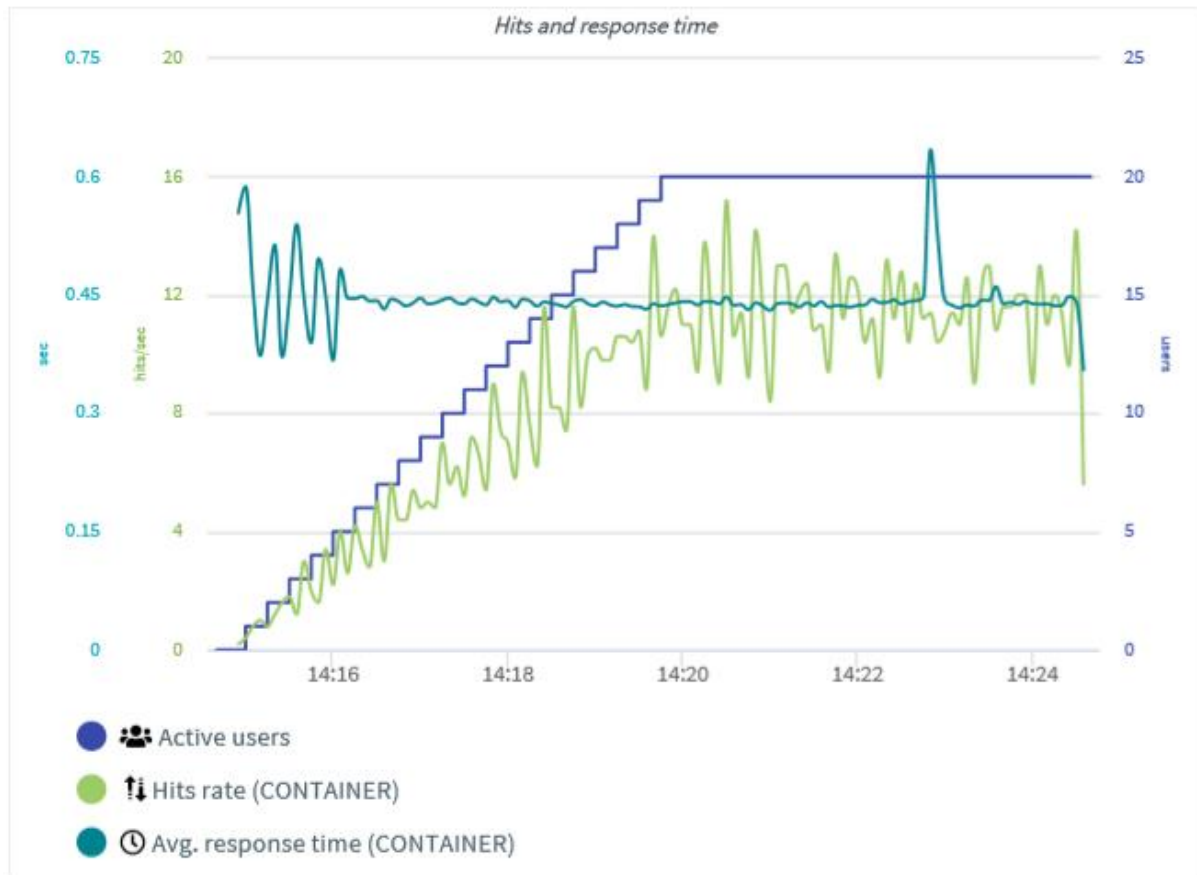
TC – Professor ImportCSV

1. Test Summary



2. Results Summary

2.1. Hit rate and response time under load











2.2. Request details




<i>Results tree</i>					
	Action	Avg. response time (sec)	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
	 AS RECORDED - T...				
	 TC - Professor ...	0.453	0.456	1266	0
	 TC - Professor ...	0.432	0.442	1266	0
	 TC - Professor ...	0.655	0.666	1250	0
	 TC - Professor ...	0.235	0.238	1250	0

Total Items: 14 (Showing Items: 5)

-  Avg. response time
-  Response time percentile 90
-  Hits count
-  Errors count

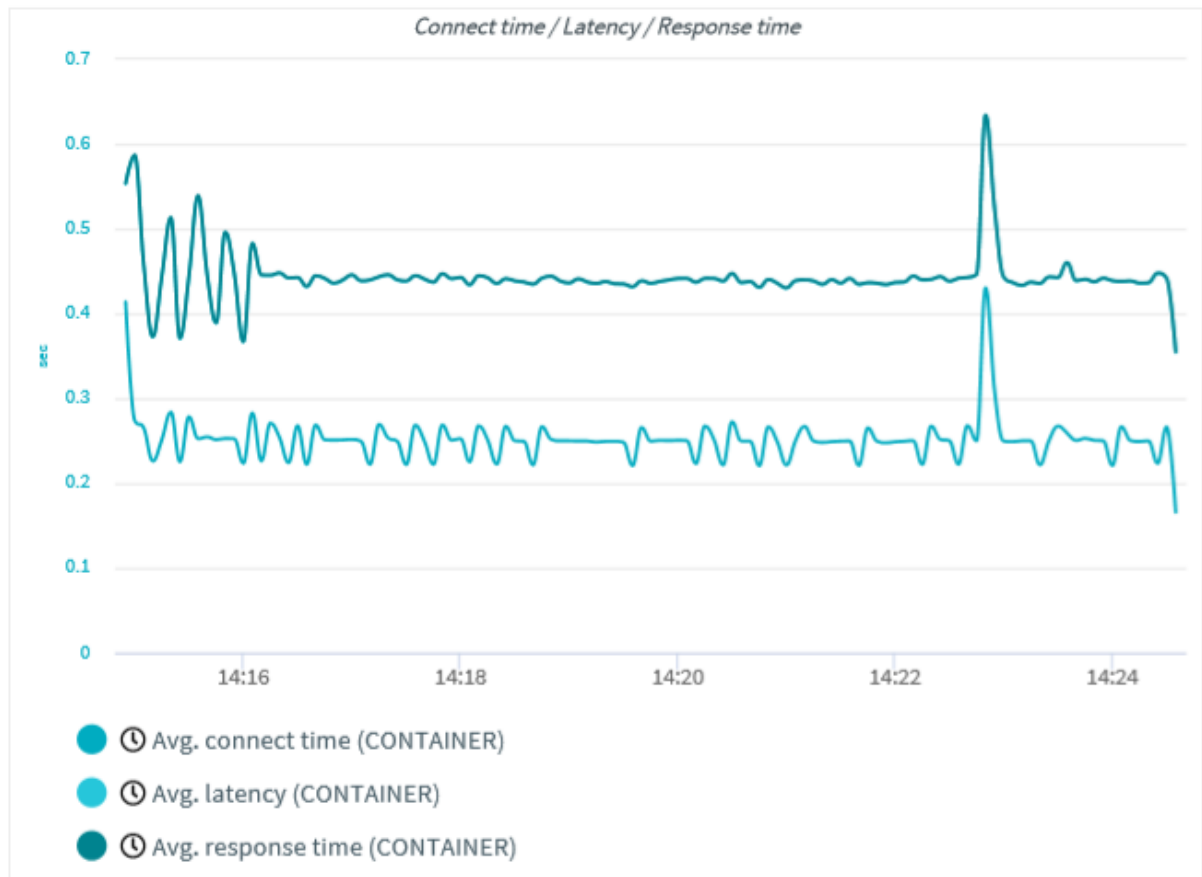
Action	Results			
	Avg. response time (sec) ▾	Response time percentile 90 (sec)	Hits count (hits)	Errors count (errors)
 /sql_atalho.php	0.276	0.285	1266	0
 /sql_logoff.php	0.235	0.238	1250	0
 /sql_professor.php?...	0.234	0.237	1266	0
 /sql_janelas.php	0.232	0.232	1266	0
 /	0.222	0.224	1266	0
 /CadAdm/cadUsuari...	0.221	0.22	1250	0
 /CadAdm/cadUsuari...	0.2	0.209	1250	0
 /sql_atalho.php	0.09	0.095	1266	0
 /sql_professor.php	0.066	0.067	1266	0

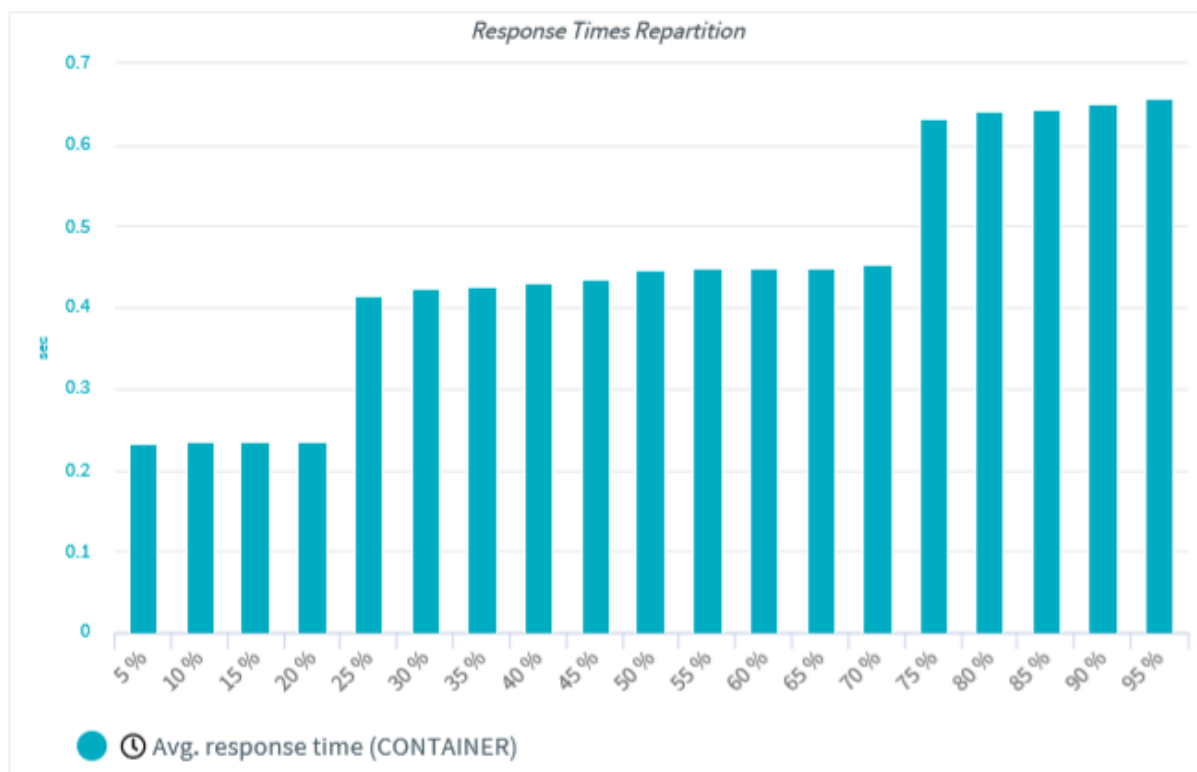
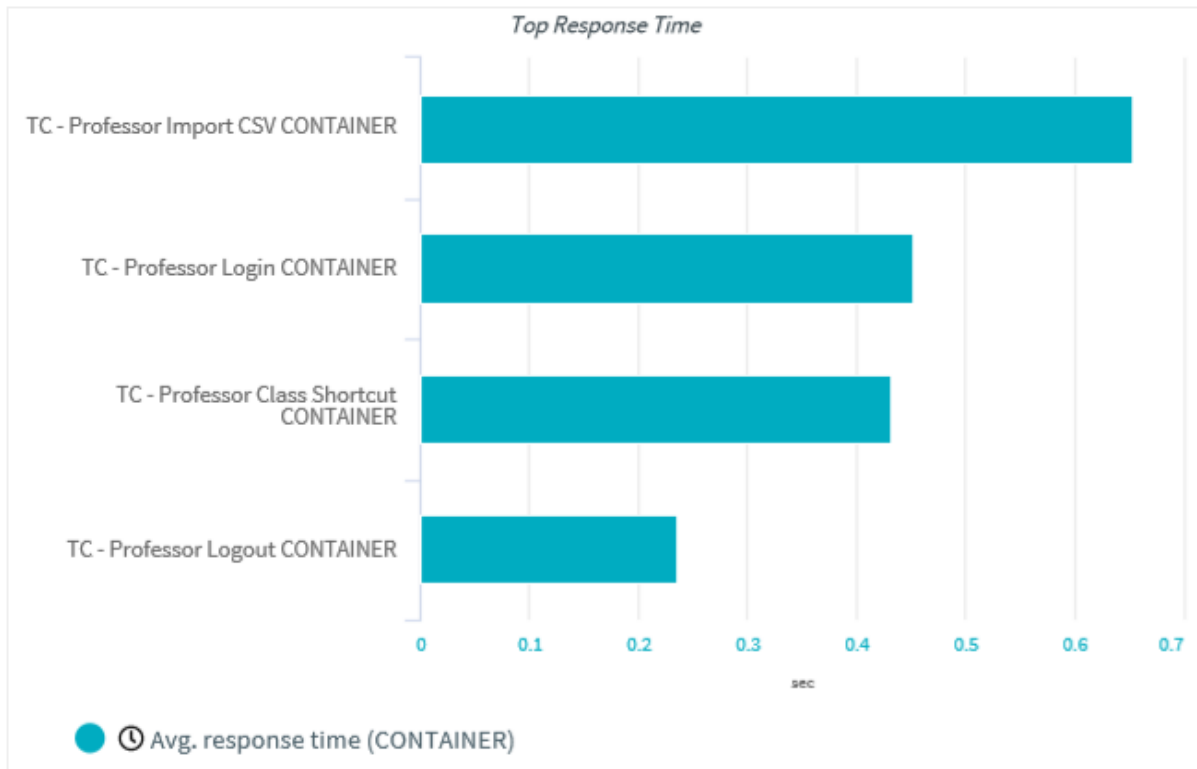
Total Items: 9

-  ⌚ Avg. response time
-  ⌚ Response time percentile 90
-  📈 Hits count
-  ❗ Errors count

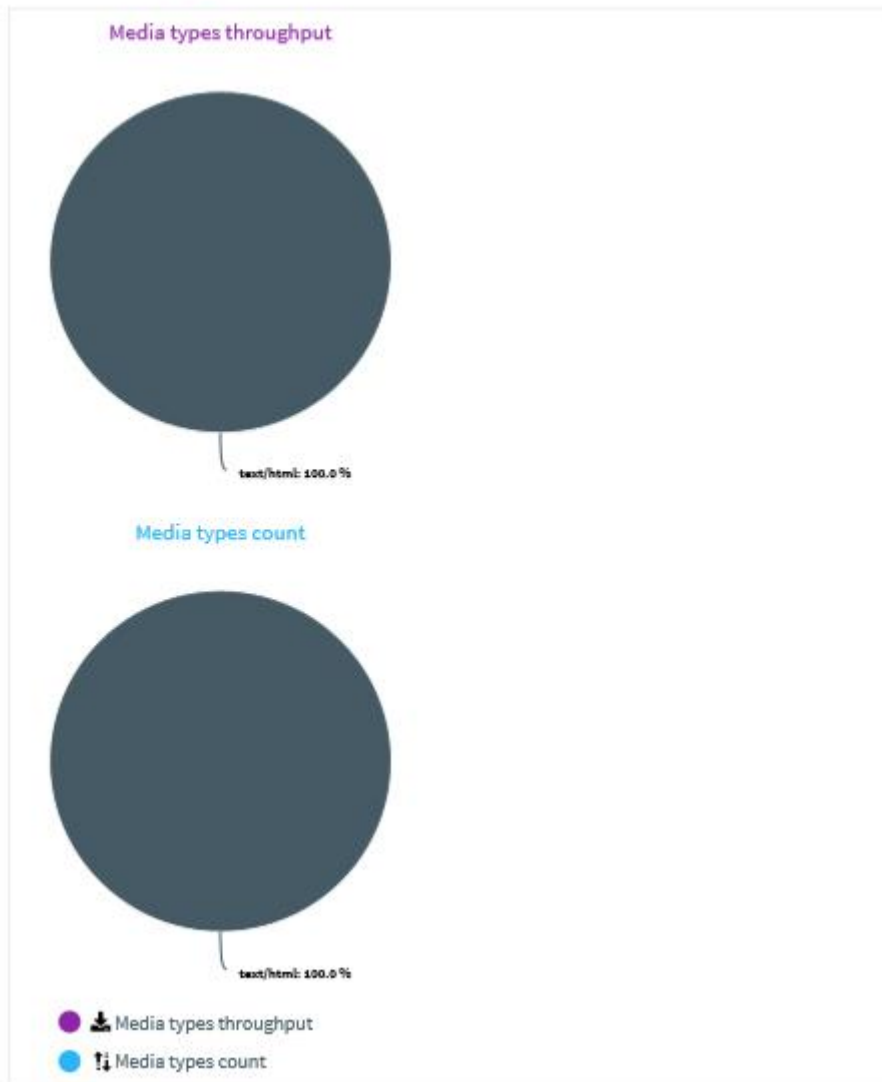
3. Result breakdown

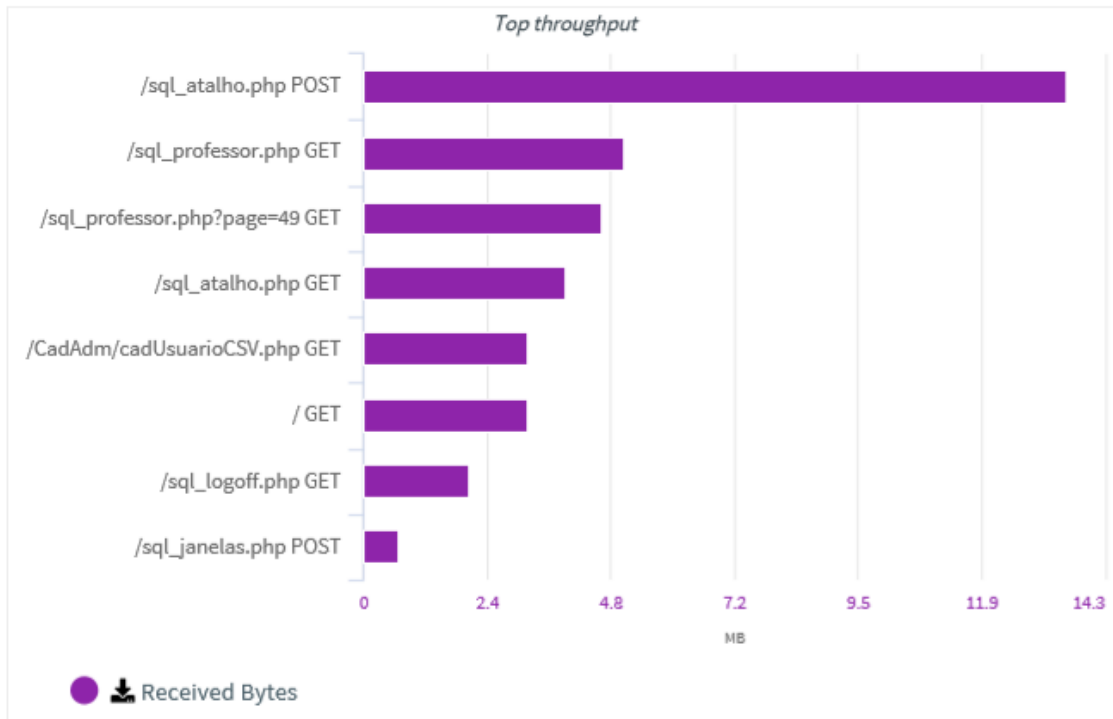
3.1. Response time and percentiles





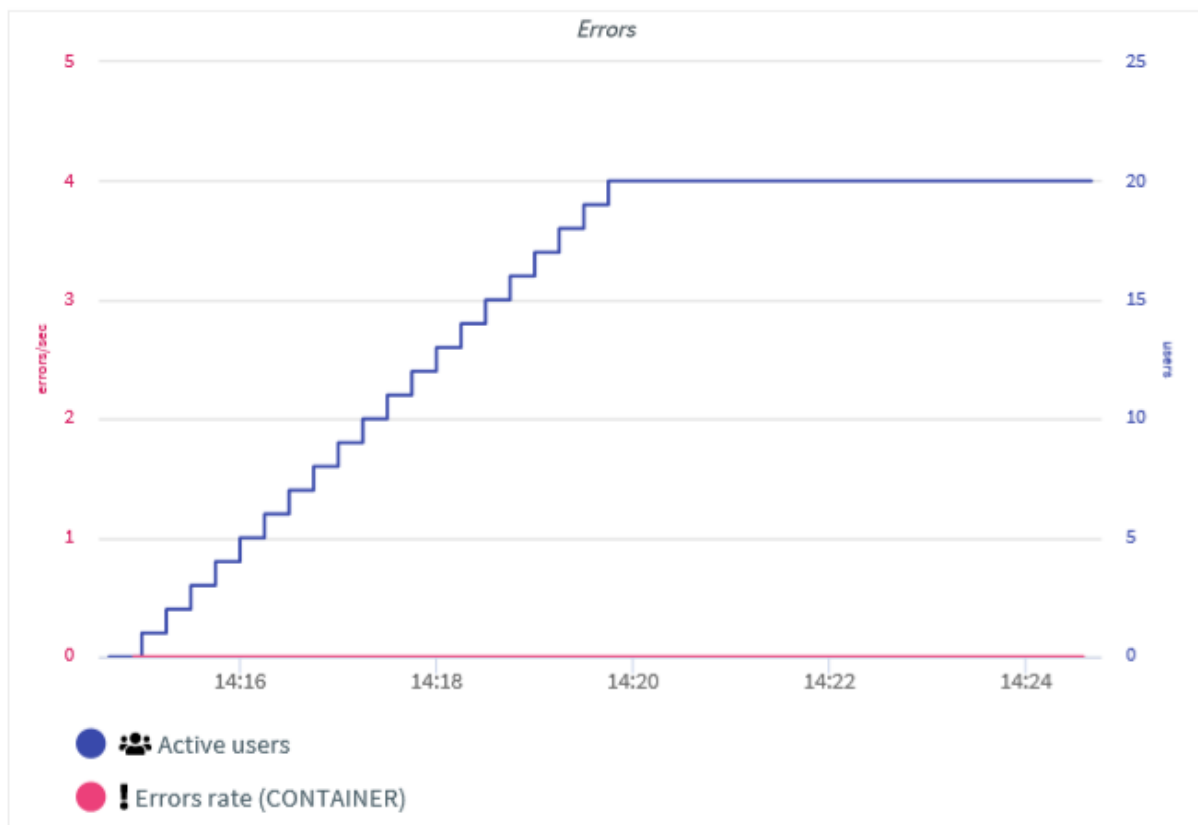
3.2. Throughput

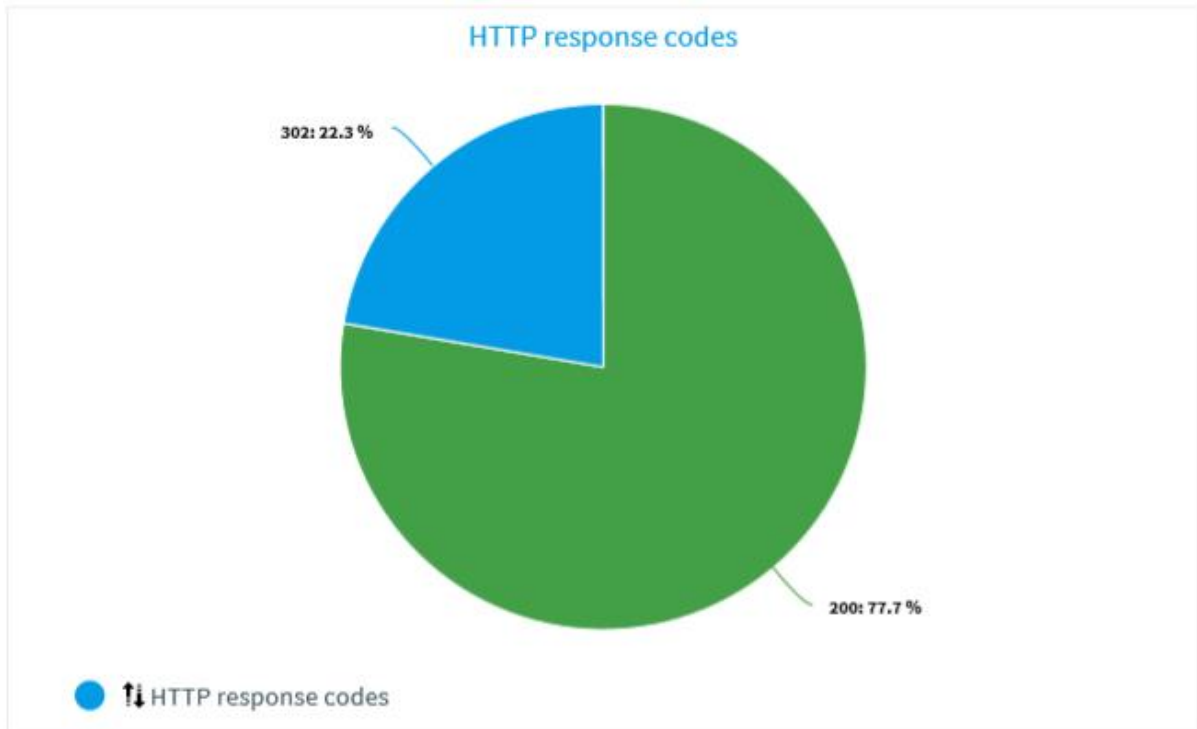




4. Error Details

This chapter presents the error rate and detailed errors.





<i>Errors</i>		
Time ▲	Action	Error
	<input type="text"/>	<input type="text"/>
2018-12-23, 14:24:44	↓/CadAdm/cadUsuariImportCSV.php	HTTP Status code: Non HTTP response code:...
2018-12-23, 14:24:44	↓/sql_atalho.php	HTTP Status code: Non HTTP response code:...
Total Items: 2		

A OctoPerf disponibiliza um glossário de terminologias utilizadas no final dos relatórios. O glossário está abaixo para consulta.

Glossary

- **Active users/User Load:** Number of active users as defined in the load test scenario.
- **APDEX (Application Performance Index):** It is a standard method for reporting performance of applications. Based on 2 thresholds, a value between 0 and 1 is computed where 0 stands for 0 users satisfied, and 1 for all users satisfied.
- **Assertions:** Count of assertions in error, failed, or successful. Assertions in error or failed lets you know that your servers did not answer as you expected.
- **Connect Time:** Time between the request and the server connection. This represents the time it took to establish the connection, including SSL handshake.
- **Containers:** Logical action which lets you group a set of requests. Usually containers correspond to a screen or web page for a better readability of the results.
- **Errors:** Count or rate of errors that occurred. Errors may happen if your virtual user is not properly designed. Otherwise, errors may be the sign that your servers are overloaded.
- **Hits:** Count or rate of hits (requests) that occurred. Hits indicate the level of load simulated to the server during the test. They should be inversely proportional to the response time.
- **HTTP Response code:** Code sent by the server indicating the status of the response to an http request. As a general rule, codes such as 1XX, 2XX and 3XX indicate a success whereas codes starting with 4XX or 5XX indicate a failure.
- **Latency:** Time between the request and the first response byte. This measures the latency from just before sending the request to just after the first response has been received. This includes all the processing to assemble the request as well as the first part of the response.
- **Percentile:** A percentile (or a centile) is a measure used in statistics indicating the value below which a given percentage of observations in a group of observations fall.
- **Response time:** Time between the request and the end of the response. Also called server response time because it does not includes the client rendering time. The response time includes both the latency and the connect time.
- **Standard deviation:** Simply the square root of the variance. It's easier to compare to other metric types using a common unit.
- **Throughput:** Bit rate in Bytes per second. Amount of data exchanged between the simulated clients and the servers.
- **Variance:** The variance quantifies the dispersion of the metric. A variance close to 0 indicates that the metric values tend to be very close to the mean, while a high variance indicates that the values are spread out over a wider range. Its unit is the square of the metric unit.

Anexo C: Caso de Teste - Perfil Aluno

ID	Caso de Teste	Localização	Descrição	Pré-condição	Procedimento	Resultado Esperado	Data de Teste	Resultado
1. Funções Gerais								
TC 45	Log in	http://labder.d ei.uc.pt/	Verifica se o aluno consegue entrar no sistema através do seu login e senha.	O login e senha já devem ter sido registados pelo administrador do sistema	1: Entre com o login 2: Entre com a senha 3: Escolha a Língua	Se for o primeiro acesso do aluno abre então a janela de consentimento informado. Se for o segundo acesso do aluno e já existir uma avaliação ativa, abre então a janela com a lista de avaliações disponíveis. Se for o segundo acesso do aluno e não existir avaliação ativa, abre então a janela de visualização de conteúdo.	02/set	Passou
TC 46	Log out	http://labder.d ei.uc.pt/	Verifica se o aluno consegue sair do sistema.	O aluno está logado	1: Clica em "Sair" 2: Clica no Botão "Sair" na janela de confirmação	Log off efetuado com Sucesso!	02/set	Passou
TC 57	Ver Conteúdo	http://labder.d ei.uc.pt/sql_m ain.php	Permite ao aluno visualizar o conteúdo de tópico de banco de dados, registado pelo professor.	O aluno está logado no painel de Menu com as principais atividades.	1: Seleciona um tópico	Mostrar o conteúdo do tópico selecionado.	02/set	Passou
TC 58	SQL - Livre	http://labder.d ei.uc.pt/sql_e exercicio_livre. php	Permite ao aluno executar comandos SQL.	O aluno está logado.	1: Clicar em "SQL - Livre" 2: Escrever um comando SQL 3: Clicar em "Enviar"	Mostra o resultado do comando SQL.	02/set	Passou
TC 83	Responder Questões de Escolha múltipla	http://labder.d ei.uc.pt/sql_a valiacao.php	Permite ao aluno responder às questões de escolha múltipla de uma avaliação	O aluno está logado e existe pelo menos uma lista de avaliação registada pelo professor	1: Selecionar uma avaliação 2: Selecionar uma questão 3: Se a questão for de escolha múltipla, seleciona uma opção de resposta e clique em "Enviar".	Apresenta a mensagem "Solução Correta - Continue assim".	02/set	Passou

TC 84	Responder a Questões de texto livres	http://labder.d ei.uc.pt/sql_a valiacao.php	Permite ao aluno responder às questões de texto livre de uma avaliação	O aluno está logado e existe pelo menos uma lista de avaliação registada pelo professor	1: Selecionar uma avaliação 2: Selecionar uma questão 3: Se a questão for de texto livre, escreva a solução e clique em "Enviar".	Resposta salva no banco de dados.	02/set	Passou
TC 85	Responder Questões de SQL	http://labder.d ei.uc.pt/sql_a valiacao.php	Permite ao aluno responder às questões de escolha múltipla de uma avaliação	O aluno está logado e existe pelo menos uma lista de avaliação registada pelo professor	1: Selecionar uma avaliação 2: Selecionar uma questão 3: Se a questão for de SQL, desenvolva o SQL e clique em "Enviar".	Apresenta a mensagem "Solução ideal 100,00% correto." e mostra o resultado do comando SQL.	02/set	Passou
TC 86	Responder Questões de ER	http://labder.d ei.uc.pt/sql_a valiacao.php	Permite ao aluno responder às questões de uma avaliação	O aluno está logado e existe pelo menos uma lista de avaliação registada pelo professor	1: Selecionar uma avaliação 2: Selecionar uma questão 3: Se a questão for de ER, desenvolva o ER e clique em "Enviar".	Resposta salva no banco de dados.	03/set	Passou
2. Gerir Aluno								
TC 47	Gerir Aluno	http://labder.d ei.uc.pt/sql_al uno.php?page =15	Permite ao aluno visualizar e editar suas informações.	Aluno está logado no painel de Menu com as principais atividades	1: Seleciona seu nome 2: Clica em "Editar" 3: Altera os campos de interesse 4: "Clica em Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e janela com a lista de alunos já registados.	03/set	Passou
TC 48	Verificar BD	http://labder.d ei.uc.pt/sql_al uno.php?page =13	Verifica se o banco de dados do aluno foi criado, podendo recriar a BD, a partir da imagem.	O aluno está logado no painel de Menu com as principais atividades	1: Seleciona "BD Criados" no campo Utilizadores 2: Seleciona Utilizador 3: Clica em "Salvar"	Apresenta lista de utilizador, onde mostra o utilizador selecionado com a BD criado.	03/set	Passou
TC 49	Frequência por mês	http://labder.d ei.uc.pt/sql_al uno.php?page =31	Permite ao aluno visualizar um relatório, com a sua frequência, durante determinado mês.	O aluno está logado no painel de Menu com as principais atividades.	1: Seleciona Mês 2: Clica "Selecionar"	Apresenta a janela com o número de frequência durante o mês especificado.	03/set	Passou
TC 50	Acessos por Utilizador - Gráfico de Colunas	http://labder.d ei.uc.pt/sql_al uno.php?page =19	Permite ao aluno visualizar um gráfico de colunas com a quantidade de acesso ao sistema.	O aluno está logado no painel de Menu com as principais atividades.	1: Clica em "Acessos por Utilizador - Gráfico de Colunas"	Apresenta a janela com gráfico de colunas que representa o número de acesso de todos os alunos.	03/set	Passou
3. Acompanhar Avaliação								

TC 51	Relatório Analisar Resultados	- http://labder.d ei.uc.pt/sql_al uno.php?page =1	Permite ao aluno visualizar um relatório com a pontuação de cada questão de uma avaliação.	O aluno está logado no painel de Menu com as principais atividades. Uma avaliação foi aplicada.	1: Seleciona uma avaliação 2: Clica em "Selecionar"	Apresenta a janela com a pontuação e número de tentativas, para cada questão, de uma determinada avaliação.	03/set	Passou
TC 52	Relatório Avaliações com notas	- http://labder.d ei.uc.pt/sql_al uno.php?page =32	Permite ao aluno visualizar um relatório com a nota de uma determinada avaliação.	O aluno está logado no painel de Menu com as principais atividades.	1: Seleciona uma ou várias avaliações 2: Clica em "Selecionar"	Apresenta a janela com a nota de uma ou mais avaliações.	04/set	Passou
TC 53	Relatório Desempenho do utilizador	- http://labder.d ei.uc.pt/sql_al uno.php?page =18	Permite ao aluno acompanhar a evolução das suas avaliações.	O aluno está logado no painel de Menu com as principais atividades.	1: Seleciona avaliação 2: Clica em "Selecionar"	Apresenta a janela com o gráfico de percentagem de andamento de uma determinada avaliação.	04/set	Passou
TC 54	Visualizar Avaliações	http://labder.d ei.uc.pt/sql_a valiacao.php	Permite ao aluno visualizar a lista de avaliações ativas.	O aluno está logado no painel de Menu com as principais atividades. Existem uma ou mais avaliações ativas.	1: Clica em "Visualizar avaliações"	Apresenta a janela com a lista de avaliações ou exercícios ativados pelo professor.	04/set	Passou
4. Explorar Conteúdo								
TC 55	Exportar Questões PDF	http://labder.d ei.uc.pt/sql_al uno.php?page =26	Permite ao aluno exportar todas as questões registadas.	O aluno está logado no painel de Menu com as principais atividades. A questão deve estar registada.	1: Clicar em "Fazer Download"	O arquivo pdf é descarregado.	04/set	Passou
TC 56	Material de Apoio	http://labder.d ei.uc.pt/sql_al uno.php?page =34	Permite ao aluno visualizar uma lista de materiais de apoio para download.	O aluno está logado no painel de Menu com as principais atividades.	1: Clica em "Material de Apoio"	Apresentar a janela de material de apoio com a lista de itens para download	04/set	Passou
4. DER – Livre								
TC 59	Criar Tabela com click (Botão esquerdo)	http://labder.d ei.uc.pt/sqlscr ipterm/index. html	Permite ao aluno criar uma tabela no espaço de desenho	O aluno está logado no painel de desenho ER.	1: Clicar em "Add Table" 2: Digita nome 3: Clica "OK"		05/set	Passou
TC 60	Editar Tabela com click (Botão esquerdo)		Permite ao aluno editar uma tabela no espaço de desenho	Pelo menos uma tabela deve estar criada.	1: Seleciona uma tabela 2: Clica em "Edit" no menu 3: Faz alterações 4: Clica "OK"		05/set	Passou

TC 61	Excluir Tabela - com click (Botão esquerdo)		Permite ao aluno criar excluir uma tabela no espaço de desenho	Pelo menos uma tabela deve estar criada.	1: Seleciona uma tabela 2: Clica em "Remove" no menu 4: Clica "OK"		05/set	Passou
TC 62	Criar Atributo - com click (Botão esquerdo)		Permite ao aluno criar um atributo no espaço de desenho	Pelo menos uma tabela deve estar criada.	1: Seleciona tabela 2: Clica "Add" no menu 3: Digita nome do atributo 4: Clica em "Save"		05/set	Passou
TC 63	Editar Atributo - com click (Botão esquerdo)		Permite ao aluno editar um atributo no espaço de desenho	Pelo menos um atributo deve estar criado.	1: Seleciona atributo 2: Clica em "Edit" no menu 3: Edita campos de interesse 4: Clica em "Save"		05/set	Passou
TC 64	Excluir Atributo - com click (Botão esquerdo)		Permite ao aluno excluir um atributo no espaço de desenho	Pelo menos um atributo deve estar criado.	1: Seleciona atributo 2: Clica em "Remove" no menu 3: Clica "OK"		05/set	Passou
TC 65	Criar Relacionamento - com click (Botão esquerdo)		Permite ao aluno criar um relacionamento no espaço de desenho	Pelo menos duas tabelas devem estar criadas.	1: Seleciona o atributo 2: Clica em "Connect FKKey" no menu 3: Clica no atributo da 2 tabela		05/set	Passou
TC 66	Editar Relacionamento - com click (Botão esquerdo)		Permite ao aluno editar um relacionamento no espaço de desenho	Pelo menos um relacionamento deve estar criado.	1: Seleciona relacionamento 2: Clica em "Edit Relation(R)" no menu 3: Altera tipo de relacionamento 4: Clica "Ok"		05/set	Passou
TC 67	Excluir Relacionamento - com click (Botão esquerdo)		Permite ao aluno excluir um relacionamento no espaço de desenho	Pelo menos um relacionamento deve estar criado.	1: Seleciona relacionamento 2: Clica em "Remove FKKey" no menu		05/set	Passou
TC 68	Adicionar PK - com click (Botão esquerdo)		Permite ao aluno adicionar uma chave primária	Pelo menos um atributo deve estar criado.	1: Seleciona atributo 2: Clica em "PK(P)" no menu		05/set	Passou
TC 69	Remover PK - com click (Botão esquerdo)		Permite ao aluno remover uma chave primária	Pelo menos um atributo PK deve estar criado.	1: Selecione o atributo PK 2: Clica em "PK(P)" no menu		05/set	Passou
TC 70	Habilitar Unique - com click (Botão esquerdo)		Permite ao aluno habilitar um atributo como único.	Pelo menos um atributo deve estar criado.	1: Seleciona o atributo 2: Clica em "Unique (N)" no menu		05/set	Passou

TC 71	Desabilitar Unique - com click (Botão esquerdo)		Permite ao aluno desabilitar um atributo como único.	Pelo menos um atributo único deve estar criado.	1: Seleciona o atributo Unique 2: Clica em "Unique (N)" no menu		05/set	Passou
TC 72	Criar Tabela - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno criar uma tabela no espaço de desenho	O aluno está logado no painel de desenho ER.	1: Clicar com lado direito no painel desenho 2: Clica em "Add new table" 3: Digita nome 4: Clica "OK"		05/set	Passou
TC 73	Editar Tabela - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno editar uma tabela no espaço de desenho	Pelo menos uma tabela deve estar criada.	1: Seleciona uma tabela 2: Clica com lado direito do rato em "Edit" 3: Faz alterações 4: Clica "OK"		05/set	Passou
TC 74	Excluir Tabela - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno criar excluir uma tabela no espaço de desenho	Pelo menos uma tabela deve estar criada.	1: Seleciona uma tabela 2: Clica com lado direito do rato em "Delete" 3: Clica "OK"		05/set	Passou
TC 75	Criar Atributo - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno criar um atributo no espaço de desenho	Pelo menos uma tabela deve estar criada.	1: Seleciona uma tabela 2: Clica com lado direito do rato em "Add new row" 3: Digita nome do atributo 4: Clica "OK"		05/set	Passou
TC 76	Editar Atributo - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno editar um atributo no espaço de desenho	Pelo menos um atributo deve estar criado.	1: Seleciona um atributo 2: Clica com lado direito do rato em "Edit" 3: Altera campos de interesse 4: Clica "OK"		05/set	Passou
TC 77	Excluir Atributo - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno excluir um atributo no espaço de desenho	Pelo menos um atributo deve estar criado.	1: Seleciona um atributo 2: Clica com lado direito do rato em "Delete" 3: Clica "OK"		05/set	Passou
TC 78	Criar Relacionamento - click com Direito Finaliza com		Permite ao aluno criar um relacionamento no espaço de desenho	Pelo menos duas tabelas devem estar criadas.	1: Seleciona chave primaria 2: Clica com lado direito do rato em "Create Foreign key"		05/set	Passou

	click (Botão esquerdo)				3: Clica na tabela onde será criada			
TC 79	Editar Relacionamento - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno editar um relacionamento no espaço de desenho	Pelo menos um relacionamento deve estar criado.	1: Seleciona um relacionamento 2: Clica com lado direito do rato em "Edit" 3: Altera tipo de relacionamento 4: Clica "OK"		05/set	Passou
TC 80	Excluir Relacionamento - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno excluir um relacionamento no espaço de desenho	Pelo menos um relacionamento deve estar criado.	1: Seleciona um relacionamento 2: Clica com lado direito do rato em "Delete"		05/set	Passou
TC 81	Adicionar PK - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno adicionar uma chave primária	Pelo menos um atributo deve estar criado.	1: Seleciona um atributo 2: Clica com lado direito do rato em "Primary"		05/set	Passou
TC 82	Remover PK - click com Direito Finaliza com click (Botão esquerdo)		Permite ao aluno remover uma chave primária	Pelo menos um atributo PK deve estar criado.	1: Seleciona uma chave primária 2: Clica com lado direito do rato em "Primary"		05/set	Passou

Anexo D: Caso de Teste - Perfil Professor

ID	Caso de Teste	Localização	Descrição	Pré-condição	Procedimento	Resultado Esperado	Data de Teste	Resultado
1. Funções Gerais								
TC 01	Log in	http://labder.dei.uc.pt/	Verifica se o professor consegue entrar no sistema usando os seus dados de log in e senha.	O log in e a senha já devem ter sido registados pelo administrador do sistema	1: Entre com o log in 2: Entre com a senha 3: Escolha a Língua	Se for o primeiro acesso do professor, abre então a janela de consentimento informado para o professor. Se for segundo acesso do professor, abre a janela para selecionar turma.	02/set	Passou
TC 02	Log out	http://labder.dei.uc.pt/sql_professor.php#	Permite ao professor sair do sistema.	Professor está logado no painel de Menu com as principais atividades.	1: Clique em "Sair" 2: Selecione o botão "Sair", para confirmar a ação de log out do sistema.	Apresenta mensagem "Log off efetuado com Sucesso!" e a janela com a lista de turmas registadas	02/set	Passou
TC 32	Trocar Turma	http://labder.dei.uc.pt/	Permite ao professor selecionar uma turma.	O professor deve estar logado.	1: Selecione a opção de visualizar todas as turmas ou os últimos acessos 2: Selecione uma turma 3: Clique em "Selecione uma turma"	Se for o primeiro acesso do professor, abre então a lista de todas as turmas disponíveis do professor. Se for o segundo acesso do professor, abre então uma lista das últimas turmas que foram acedidas. Após selecionar uma turma, é apresentada a janela de menu principal com as principais atividades.	02/set	Passou

TC 47	SQL-Livre	https://labder.dei.uc.pt/sql_exercicio_livre.php	Permite ao aluno executar comandos SQL.	O aluno está logado.	1: Clique em "SQL - Livre" 2: Escreva um comando SQL 3: Clique em "Enviar"	Mostra o resultado do comando SQL.	02/set	Passou
2. Gerir Alunos								
TC 03	Adicionar Aluno	http://labder.dei.uc.pt/sql_professor.php?page=14	Permite registar um novo aluno no sistema.	Professor está logado no painel de Menu com as principais atividades	1: Digite a matrícula do aluno 2: Digite o nome do aluno 3: Digite o e-mail do aluno 4: Digite o log in do aluno 5: Digite a senha do aluno	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de alunos já registados.	02/set	Passou
TC 04	Importar Aluno CSV	http://labder.dei.uc.pt/sql_professor.php?page=49	Permite importar alunos através de um arquivo CSV.	Professor está logado no painel de Menu com as principais atividades.	1: Clique em "Escolher Arquivo" 2: Clique em "Upload"	Apresenta a lista com os dados dos alunos importados.	02/set	Passou
TC 05	Gerir Aluno - Visualizar e Editar	http://labder.dei.uc.pt/sql_professor.php?page=15	Permite visualizar e editar dados dos alunos.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione o aluno, na lista de alunos 2: Clique em "Editar" 3: Altere os campos de interesse 4: "Clique em Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de alunos já registados.	02/set	Passou
TC 06	Verificar BD	http://labder.dei.uc.pt/sql_professor.php?page=13	Verifica se a BD do aluno foi criada, e permite recriar a BD a partir de uma imagem.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione "BD Criados" no campo Utilizadores 2: Selecione Utilizador 3: Clica em "Salvar"	Apresenta lista de utilizadores onde mostra o utilizador selecionado com a BD criado.	02/set	Passou
TC 07	Adicionar Equipa	http://labder.dei.uc.pt/sql_professor.php?page=23	Relacionar uma atividade a um grupo de alunos.	Professor está logado no painel de Menu com as principais atividades. A atividade deve estar registada.	1: Selecione a atividade (avaliação) 2: Digite o nome da Equipa 3: Escolha data e hora de início 4: Escolha data e hora do fim 5: Selecione participante do curso 6: Repasse para o campo "componentes da Equipa" 7: Clique em Salvar	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de alunos por Equipa.	02/set	Passou
TC 08	Gerir Equipa	http://labder.dei.uc.pt/sql	Permite editar dados da Equipa.	Professor está logado no painel de Menu com as	1: Selecione a Equipa 2: Clique em "Editar"	Apresenta mensagem "A operação foi	02/set	Passou

		_professor.php?page=24		principais atividades. A Equipa deve estar registada.	3: Altere campos de interesse 4: Clique em " Salvar"	realizada com sucesso!" e a janela com a lista das Equipas registadas.		
TC 09	Gerir Alunos da Equipa	http://labder.dei.uc.pt/sql_professor.php?page=25	Permite visualizar os dados dos alunos que compõem as Equipas.	Professor está logado no painel de Menu com as principais atividades. A Equipa deve estar registada.	1: Selecione aluno 2: Clique em "Detalhes"	Apresenta a janela com a Equipa e o nome do aluno.	02/set	Passou
3. Gerir Avaliações								
TC 10	Adicionar Avaliação	http://labder.dei.uc.pt/sql_professor.php?page=2	Permite ao professor criar uma frequência.	Professor está logado no painel de Menu com as principais atividades. As questões devem estar registadas e agrupadas.	1: Selecione Grupo de questões 2: Digite a quantidade do problema 3: Selecione a data e hora do início 4: Selecione a data e hora do término 5: Digite o valor da avaliação 6: Selecione a situação da avaliação 7: Digite a descrição da avaliação 8: Clique em "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de avaliações registadas.	02/set	Passou
TC 11	Gerir Avaliação	http://labder.dei.uc.pt/sql_professor.php?page=3	Permite ao professor editar dados da avaliação.	Professor está logado no painel de Menu com as principais atividades. A avaliação deve estar registada.	1: Selecione uma avaliação 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de avaliações registadas.	02/set	Passou
TC 12	Agrupar Questões	http://labder.dei.uc.pt/sql_professor.php?page=11	Permite ao professor associar as questões a um grupo.	Professor está logado no painel de Menu com as principais atividades. As questões devem estar registadas. O grupo deve estar registado.	1: Selecione Grupo 2: Selecciona Questão 3. Clique em "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de grupos registados.	02/set	Passou
TC 13	Gerir Grupo de Questões	http://labder.dei.uc.pt/sql	Permite ao professor editar dados do grupo.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione o grupo 2: Clique em "Editar"	Apresenta mensagem "A operação foi realizada com	02/set	Passou

		_professor.php?page=5		O grupo de questões deve estar registado.	3: Altere os campos de interesse 4: Clique em "Salvar"	sucesso!" e a janela com a lista de grupos registados.		
TC 14	Liberar Avaliação	http://labder.dei.uc.pt/sql_professor.php?page=6	Permite ao professor ativar ou desativar uma frequência ou exercício.	Professor está logado no painel de Menu com as principais atividades. A avaliação deve estar registada.	1: Selecione a situação da avaliação 2: Selecione a avaliação 3: Selecione os alunos 4: Clique em "Alterar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de avaliações registados.	02/set	Passou
TC 15	Relatório - Analisar Resultados	http://labder.dei.uc.pt/sql_professor.php?page=1	Permite ao professor visualizar um relatório com a pontuação das avaliações por aluno.	Professor está logado no painel de Menu com as principais atividades. Uma avaliação foi aplicada.	1: Selecione uma avaliação 2: Clique em "Selecionar" 3: Clique na nota que deseja alterar 4: Selecione a questão 5: Digite a nova nota 6: Clique em "Alterar Nota"	Apresenta a mensagem "Alterou a nota para:" e a nova nota.	02/set	Passou
TC 16	Relatório - Avaliações com notas	http://labder.dei.uc.pt/sql_professor.php?page=32	Permite ao professor visualizar um relatório com a nota das avaliações de todos os alunos.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione uma ou várias avaliações 2: Clique em "Selecionar"	Apresenta a janela com a lista de alunos e suas respectivas notas, por avaliação.	02/set	Passou
TC 17	Relatório - Acompanhar Avaliação	http://labder.dei.uc.pt/sql_professor.php?page=18	Permite ao professor acompanhar o andamento das avaliações.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione avaliação 2: Clique em "Selecionar"	Apresenta a janela com gráfico de percentagem de andamento da avaliação de todos os alunos.	02/set	Passou
4. Acompanhar Acesso								
TC 18	Frequência por mês	http://labder.dei.uc.pt/sql_professor.php?page=31	Permite ao professor visualizar um relatório com a frequência por aluno durante determinado mês.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione Mês 2: Clique "Selecionar"	Apresenta a janela com o número de frequência por aluno, durante o mês especificado.	03/set	Passou
TC 19	Acessos por Utilizador - Gráfico de Barras	http://labder.dei.uc.pt/sql_professor.php?page=17	Permite ao professor visualizar o gráfico de barras com o número de acessos por alunos.	Professor está logado no painel de Menu com as principais atividades.	1: Clique em "Acessos por Utilizador - Gráfico de Barras"	Apresenta a janela com gráfico de barras que representa o número de acesso de todos os alunos.	03/set	Passou

TC 20	Acessos por Utilizador - Gráfico de Colunas	http://labder.dei.uc.pt/sql_professor.php?page=19	Permite ao professor visualizar o gráfico de colunas com o número de acessos por alunos.	Professor está logado no painel de Menu com as principais atividades.	1: Clique em "Acessos por Utilizador - Gráfico de Colunas"	Apresenta a janela com gráfico de colunas que representa o número de acesso de todos os alunos.	03/set	Passou
TC 21	Utilizadores Online	http://labder.dei.uc.pt/sql_professor.php?page=28	Permite ao professor visualizar quais alunos estão online e offline.	Professor está logado no painel de Menu com as principais atividades.	1: Clique em "Utilizadores Online"	Apresenta a janela com lista de utilizador e sua respetiva situação (online ou offline).	03/set	Passou
5. Gerir Questões								
TC 22	Adicionar Questão	http://labder.dei.uc.pt/sql_professor.php?page=7	Permite ao professor registar uma questão.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione o nível de dificuldade 2: Digite o enunciado da questão 3: Selecione tipo de problema 4: Digite o número de tentativas 5: Adicione imagem 6: Clica em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de questões registadas.	03/set	Passou
TC 23	Gerir Questão	http://labder.dei.uc.pt/sql_professor.php?page=8	Permite ao professor editar dados da questão.	Professor está logado no painel de Menu com as principais atividades. A questão deve estar registada.	1: Selecione a questão 2: Clique em "Editar" 3: Altere campos de interesse 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de questões registadas.	03/set	Passou
TC 24	Adicionar Questão de Escolha múltipla	http://labder.dei.uc.pt/sql_professor.php?page=9	Permite ao professor registar uma questão de escolha múltipla.	Professor está logado no painel de Menu com as principais atividades. A questão deve estar registada.	1: Selecione Questão 2: Adicione o código de alternativa 3: Descreva as alternativas 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de questões de escolha múltipla registadas.	03/set	Passou
TC 25	Gerir Questão de Escolha múltipla	http://labder.dei.uc.pt/sql_professor.php?page=10	Permite ao professor alterar os dados de uma questão de escolha múltipla.	Professor está logado no painel de Menu com as principais atividades. A questão de escolha múltipla deve estar registada.	1: Selecione a questão 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de questões de escolha múltipla registadas.	03/set	Passou

TC 26	Adicionar Grupo	http://labder.dei.uc.pt/sql_professor.php?page=11	Permite ao professor registar o nome de um grupo.	Professor está logado no painel de Menu com as principais atividades.	1: Digite descrição do grupo 2: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de grupos registados.	03/set	Passou
TC 27	Gerir Grupo	http://labder.dei.uc.pt/sql_professor.php?page=12	Permite ao professor alterar o nome do grupo.	Professor está logado no painel de Menu com as principais atividades. O grupo deve estar registado.	1: Selecione o grupo 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de grupos registados.	03/set	Passou
TC 28	Adicionar Respostas	http://labder.dei.uc.pt/sql_professor.php?page=21	Permite ao professor registar uma resposta para uma questão.	Professor está logado no painel de Menu com as principais atividades. A questão deve estar registada.	1: Selecione questão 2: Digite a descrição da resposta 3: Digite o número de pontos da resposta 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de questões com respostas registadas.	03/set	Passou
TC 29	Gerir Respostas	http://labder.dei.uc.pt/sql_professor.php?page=22	Permite ao professor alterar a resposta de uma questão.	Professor está logado no painel de Menu com as principais atividades. A questão deve estar registada. A resposta deve estar registada.	1: Selecione a Questão 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de questões com respostas registadas.	03/set	Passou
TC 30	Exportar Questões (PDF)	http://labder.dei.uc.pt/sql_professor.php?page=26	Permite ao professor exportar todas as questões registadas.	Professor está logado no painel de Menu com as principais atividades. A questão deve estar registada.	1: Clique em "Fazer Download"	O arquivo PDF é descarregado.	03/set	Passou
TC 31	Exportar Questões com Respostas (PDF)	http://labder.dei.uc.pt/sql_professor.php?page=33	Permite ao professor exportar todas as questões com resposta.	Professor está logado no painel de Menu com as principais atividades. A questão deve estar registada. A resposta deve estar registada,	1: Clique em "Fazer Download"	O arquivo PDF é descarregado.	03/set	Passou

5. Gerir Departamento								
TC 33	Adicionar Professor	http://labder.dei.uc.pt/sql_professor.php?page=47	Permite adicionar um novo professor para aceder ao ambiente de trabalho.	Professor está logado no painel de Menu com as principais atividades.	1: Digite código 2: Digite nome 3: Digite e.mail 4: Digite login 5: Digite senha 6: Clique em "salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de professores registados.	04/set	Passou
TC 34	Gerir Professor	http://labder.dei.uc.pt/sql_professor.php?page=48	Permite ao professor alterar dados de um professor.	Professor está logado no painel de Menu com as principais atividades. O professor deve estar registado.	1: Selecione professor 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de professores registados.	04/set	Passou
TC 35	Adicionar Instituição	http://labder.dei.uc.pt/sql_professor.php?page=45	Permite ao professor registar uma nova instituição.	Professor está logado no painel de Menu com as principais atividades.	1: Digite código da instituição 2: Digite nome da instituição 3: Digite informação da instituição 4: Digite o link da instituição 5: Clique em "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de instituições registadas.	04/set	Passou
TC 36	Gerir Instituição	http://labder.dei.uc.pt/sql_professor.php?page=46	Permite ao professor alterar os dados de uma instituição.	Professor está logado no painel de Menu com as principais atividades. A instituição deve estar registada.	1: Selecione instituição 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de instituições registadas.	04/set	Passou
TC 37	Adicionar Tipo de Curso	http://labder.dei.uc.pt/sql_professor.php?page=43	Permite ao professor registar um tipo de curso.	Professor está logado no painel de Menu com as principais atividades. A instituição deve estar registada.	1: Selecione instituição 2: Digite o código do tipo do curso 3: Digite a descrição 4: Digite as informações 5: Clique em "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de tipo de cursos registados.	04/set	Passou
TC 38	Gerir Tipo de Curso	http://labder.dei.uc.pt/sql_professor.php?page=44	Permite ao professor alterar dados de um tipo de curso.	Professor está logado no painel de Menu com as principais atividades. O tipo de curso deve estar registado.	1: Selecione o tipo de curso 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem " A operação foi realizada com sucesso!" e a janela com a lista de tipo de cursos registados.	04/set	Passou

TC 39	Adicionar Curso	http://labder.dei.uc.pt/sql_professor.php?page=41	Permite ao professor registar um curso.	Professor está logado no painel de Menu com as principais atividades. A instituição deve estar registada. O tipo de curso deve estar registado.	1: Selecione a instituição 2: Selecione o tipo de curso 3: Digite o código do curso 4: Digite nome do curso 5: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de cursos registados.	04/set	Passou
TC 40	Gerir Curso	http://labder.dei.uc.pt/sql_professor.php?page=42	Permite ao professor alterar os dados de um curso.	Professor está logado no painel de Menu com as principais atividades. O curso deve estar registado.	1: Selecione curso 2: Clique em "Editar" 3: Altere campos de interesse 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de cursos registados.	04/set	Passou
TC 41	Adicionar Disciplina	http://labder.dei.uc.pt/sql_professor.php?page=39	Permite ao professor registar uma nova disciplina.	Professor está logado no painel de Menu com as principais atividades. A instituição deve estar registada. O tipo de curso deve estar registado. O curso deve estar registado.	1: Selecione a instituição 2: Selecione o tipo de curso 3: Selecione o curso 4: Digite o código da disciplina 5: Selecione o utilizador responsável 6: Digite o nome da disciplina 7: Digite as informações 8: Digite o público alvo 9: Digite a data de inclusão 10: Selecione a situação 11: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de disciplinas registadas.	04/set	Passou
TC 42	Gerir Disciplina	http://labder.dei.uc.pt/sql_professor.php?page=40	Permite ao professor alterar os dados da disciplina.	Professor está logado no painel de Menu com as principais atividades. A disciplina deve estar registada.	1: Selecione a disciplina 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de disciplinas registadas.	04/set	Passou
TC 43	Adicionar Turma	http://labder.dei.uc.pt/sql_professor.php?page=37	Permite ao professor registar uma nova turma.	Professor está logado no painel de Menu com as principais atividades. A instituição deve estar registada. O tipo de curso deve estar registado.	1: Selecione a instituição 2: Selecione o tipo de curso 3: Selecione o curso 4: Selecione a disciplina 5: Digite o código da turma 6: Selecione o professor 7: Digite o nome da turma	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de turmas registadas.	04/set	Passou

				O curso deve estar registado. A disciplina deve estar registada.	8: Selecione a data de inscrição 9: Selecione a data do início 10: Selecione a data do fim 11: Digite as informações 12: Digite o número máximo de alunos 13: Selecione o tipo de acesso ao visitante 14: Selecione a data da inclusão 16: Selecione a situação 17: Clique em "Salvar"			
TC 44	Gerir Turma	http://labder.dei.uc.pt/sql_professor.php?page=38	Permite ao professor alterar os dados da turma.	Professor está logado no painel de Menu com as principais atividades. A turma deve estar registada.	1: Selecione uma turma 2: Clique em "Editar" 3: Altere os campos de interesse 4: Clique em "Salvar"	Apresenta mensagem "A operação foi realizada com sucesso!" e a janela com a lista de turmas registadas.	04/set	Passou
TC 45	Gerir Conteúdo	http://labder.dei.uc.pt/sql_professor.php?page=20	Permite ao professor alterar o conteúdo do tópico do banco de dados.	Professor está logado no painel de Menu com as principais atividades.	1: Selecione um tópico 2: Clique em "Exemplo/Conceito de Detalhe" 3: Altere ou adicione conteúdo 4: Clique em "Salvar"	Apresenta o conteúdo com as alterações feitas.	04/set	Passou
TC 46	Material de Apoio	http://labder.dei.uc.pt/sql_professor.php?page=34	Permite ao professor visualizar uma lista de materiais de apoio para download.	Professor está logado no painel de Menu com as principais atividades.	1: Clique em "Material de Apoio"	Apresenta uma janela com a lista de materiais para download.	04/set	Passou

Anexo E: Plano de atividades para uso do LabDER

O plano de atividades é um documento que descreve as atividades previstas para o uso do LaBDER pelas instituições de ensino. Inclui um cronograma genérico, que é adaptado pelo professor, para a execução das atividades de exercícios e avaliações, usando a plataforma.

Plano de atividades para recolher dados de diagrama de entidade e relacionamento

Adriano Del Pino Lino

Programa de Doutoramento em Ciências e Tecnologias da Informação, Universidade de Coimbra

1. Apresentação do plano

O plano de recolha de dados de diagramas de entidade e relacionamento (DER) é uma iniciativa vinculada ao projeto de doutoramento sobre a avaliação automática de DER, em ambientes virtuais de ensino e aprendizagem, inserido no programa de doutoramento em Ciências e Tecnologias da Informação da Universidade de Coimbra, e orientada pelo Professor Doutor Álvaro Manuel Reis da Rocha e pelo Professor Doutor Luís Miguel Machado Lopes Macedo.

A recolha de dados é no formato online, com o uso da ferramenta LabDER que, nesta 1ª edição, é destinada aos alunos de formação superior, nas áreas da Ciência da Informática e áreas afins. Dessa forma, a atividade de recolha de dados no LabDER, contempla as seguintes características:

- Disponibilização de um ambiente virtual de aprendizagem para ensino e aprendizagem de diagramas de entidade e relacionamento, e recursos materiais para a interação entre os participantes em tempo real;
- A atividade de recolha de dados deve ser executada num curso semipresencial;
- Oferecido a uma quantidade limitada de pessoas (máximo de 500 participantes);
- O ambiente adapta-se ao plano de ensino do professor. Ou seja, o professor define a quantidade de listas de exercícios, a quantidade de questões de cada lista, e o tempo disponível para resolução;

- O ambiente é flexível, podem ser alteradas as datas e horas das entregas de qualquer atividade prevista pelo professor;
- As atividades podem ser acompanhadas pelo professor nas atividades de laboratório presencialmente, ou à distância. O professor pode avaliar as questões até ao final do período letivo estabelecido, no cronograma de atividades;
- O material didático é disponibilizado no sistema. Cada participante pode aceder ao material de acordo com o seu ritmo, dentro de um período pré-estabelecido (com data de início e término).

2. Objetivos e metodologia

Objetivos: Proporcionar aos utilizadores, professor e alunos, um ambiente que sirva de apoio à execução de exercícios e provas sobre DER. **Recolher os dados das respostas de exercícios e provas de DER** e recolher dados sobre a avaliação do sistema.

Metodologia: Este plano de atividades pode ser inserido no plano de ensino do curso elaborado pelo professor. A ferramenta deve ser utilizada nos momentos de atividades de realização de exercícios e provas previstos, no plano do professor. O investigador tem a tarefa de registar os alunos, o professor, as questões, as listas de exercícios e as avaliações definidas pelo professor. O investigador deve fornecer um ambiente pronto a ser utilizado.

3. Definição dos utilizadores-alvo

Os utilizadores alvo são alunos e professores do ensino superior, na área de informática, na disciplina de Base de Dados.

4. Período de realização do curso e carga horária

O curso e todo material utilizado pelos participantes ficarão disponíveis no ambiente de trabalho durante o período do desenvolvimento do curso, definido pelo professor. Tem a duração máximo de um semestre letivo, acrescido de um prazo de 6 meses.

O professor define a carga horária total do curso que será dada, ao ritmo de cada um dos participantes do curso.

O plano de ensino do professor servirá como uma estimativa a ser utilizada no LabDER. Todas as atividades previstas pelo professor serão apoiadas com a intervenção do investigador. Os participantes poderão desenvolvê-las ao seu “próprio ritmo”, contando com o suporte pedagógico do investigador.

5. Cronograma do projeto

O cronograma de execução da atividade de recolha de dados de DER tem três momentos.

A primeira fase, tem como objetivo preparar o ambiente de trabalho para a utilização dos participantes. Portanto deve ser feito, com o mínimo de duas semanas de antecedência do início das atividades académicas planeadas pelo professor. Os seguintes documentos são necessários para iniciar a preparação do ambiente de trabalho: 1 - uma lista com o nome completo e e-mail

dos alunos que vão participar no projeto e as listas de exercícios com as questões e respostas; 2 - o plano de ensino do professor, contendo o período de duração das atividades com as datas de início e término; 3 - a identificação completa do professor (instituição de ensino, nome completo, nome do curso, nome da disciplina e identificação da turma).

A segunda fase do cronograma, execução das atividades, é o momento para os alunos realizarem as atividades previstas, no plano de ensino do professor, usando o LabDER. A quantidade de exercícios, bem como a duração de cada um deles estão determinadas no plano de ensino do professor. No caso de avaliações que envolvam questões de diagramas, o professor deve submeter e avaliar manualmente a resposta do aluno.

A terceira fase do cronograma, avaliação do ambiente de trabalho, é o momento em que os participantes são convidados a avaliar o sistema. Essa atividade deve ser incluída, após a última atividade prevista no plano de ensino do professor e antes do término das atividades acadêmicas.

Abaixo está o cronograma de execução do projeto.

Tabela 1. Cronograma de atividades do projeto.

#	Fase	Descrição	Período
1	Preparação do ambiente de trabalho	Registo dos utilizadores e questões. Documentos necessários. Plano de ensino do professor, lista(s) de exercícios com respostas, lista nominal dos alunos com e-mail.	2 Semanas antes do início das atividades acadêmicas.
2	Execução das atividades	Período de interação dos participantes com o LabDER – realização dos exercícios/avaliações	Período reservado pelo professor.
		Atividade 01 – Tarefa (n questões)	
		Atividade n – Tarefa (n questões)	
		Atividade final – Avaliação (n questões)	
3	Avaliação do ambiente	Período de avaliação do ambiente pelos participantes. Responder ao questionário pelo sistema, no dia definido pelo professor.	1 Dia
	Agenda de Finalização	Encerramento do curso	

6. Desenvolvimento metodológico

São disponibilizados, o manual do utilizador com ilustrações das janelas do LabDER, e orientações, passo a passo, para a execução do ambiente de trabalho. A partir da leitura dessas orientações, cada participante poderá configurar o ambiente de trabalho e fazer o login no sistema.

Será agendada uma avaliação programada, onde cada participante fará uma avaliação do sistema e autoanálise dos conhecimentos adquiridos. Essa avaliação deverá ser realizada após a conclusão das atividades de exercícios e antes da avaliação final do professor.

O investigador, profissional de informática com conhecimentos avançados em BD atuará como tutor durante o período do curso. Ele estará à disposição dos participantes 20 horas semanais para tirar as dúvidas sobre as atividades que se destinam à configuração das ferramentas do curso. Para isso, as dúvidas precisam de ser encaminhadas ao tutor via e-mail, sendo por ele respondidas, no prazo de 24 horas. O investigador também estará disponível para resoluções de problemas técnicos.

7. Avaliação

Espera-se que cada participante possa realizar 100% das atividades previstas pelo professor, bem como 100% da avaliação do sistema por ele prevista. Todas as atividades são consideradas necessárias para o desenvolvimento do projeto de forma eficaz. O professor define os critérios, de avaliação e aprovação do aluno, no seu plano de ensino. Para isso, pode usar os relatórios sintéticos do LabDER, como suporte à sua decisão.

8. Material didático envolvido

O professor deve utilizar o seu próprio material pedagógico, no entanto o ambiente de trabalho disponibiliza um material didático aos alunos e professores e pode ser adotado como material principal de um curso de BD.

9. Requisitos

O projeto de recolha de dados de atividades de Diagrama de Entidade e Relacionamento é desenvolvido de forma semipresencial, portanto, além dos recursos disponibilizados pela organização do próprio curso, outros recursos são necessários para acesso ao LabDER.

- a. Requisitos técnicos: A leitura do manual do utilizador do LabDER, para manipular o ambiente de trabalho, deve ser concluída no período de ambientação da ferramenta. O investigador está disponível para auxiliar os participantes, em qualquer eventualidade. A seguir, encontra-se a lista dos programas necessários:
 - **Navegador Google Chrome** para aceder ao curso online (**uso obrigatório**);
 - **Adobe Acrobat Reader** que permite a visualização do material didático do curso.

- b. Requisitos de acesso: Os interessados devem reunir as seguintes condições para acesso:
 - Estar devidamente identificados e autorizados pelo professor do curso;
 - Declarar que aceita o uso de dados anónimos para fins de investigação;
 - Satisfazer os requisitos técnicos;
 - Ter uma conta válida de e-mail.

A documentação, abaixo, é solicitada ao professor para fins de registo no sistema LabDER:

- Plano de ensino, contendo a identificação completa do professor e os momentos das atividades de ensino;
- Listas de exercícios e avaliações com a descrição das questões e respectivas respostas a serem aplicadas no sistema;
- Uma lista dos alunos que participarão, contendo nome completo e e-mail;
- Qualquer material adicional a ser disponibilizado aos participantes na ferramenta.

10. Exemplo de cronograma de curso

O LabDER possui material didático próprio. Abaixo está um exemplo de cronograma de curso com conteúdo programático, organizado em cinco unidades. Em cada uma delas, estão

previstas atividades com tutoriais, que dão sugestões para que o participante perceba a potencialidade que o LabDER possui.

Um tópico de DER é lecionado por semana apresentando os conceitos em sala de aula, depois em laboratório e em seguida as execuções de listas de exercícios em conjunto com os alunos. No final da segunda semana, é aplicada a primeira avaliação formal. Este fluxo segue apenas como exemplo de divisão dos conteúdos lecionados e atividades que podem ser executadas no LabDER.

Tabela 2. Cronograma do curso.

<i>Período</i>	<i>Itens</i>		<i>Carga Horária</i>
<i>1 Semana</i>	<i>Agenda de apresentação</i>	<i>Introdução (Comunidade de Aprendizagem e Glossário)</i>	
	<i>Unidade 1</i>	<i>Leitura do Módulo VI – Introdução aos conceitos de diagrama de entidade e relacionamento</i>	<i>2h</i>
		<i>Aula no laboratório</i>	<i>1h</i>
		<i>Atividade 01 – Tarefa (XX exercícios)</i>	<i>3h</i>
	<i>Total Unidade 1</i>		<i>6h</i>
<i>2 Semana</i>	<i>Unidade 2</i>	<i>Leitura do Módulo VI – Entidade e Atributos</i>	<i>3h</i>
		<i>Aula no laboratório</i>	<i>1h</i>
		<i>Atividade 02 – Tarefa (XX exercícios)</i>	<i>3h</i>
		<i>Atividade 03 – Avaliação</i>	<i>1h</i>
	<i>Total Unidade 2</i>		<i>8h</i>
<i>3 Semana</i>	<i>Unidade 3</i>	<i>Leitura do Módulo VI – Relacionamento (Parte1)</i>	<i>3h</i>
		<i>Aula no laboratório</i>	<i>1h</i>
		<i>Atividade 4 – Tarefa (XX exercícios)</i>	<i>3h</i>
		<i>Atividade 5 – Avaliação</i>	<i>1h</i>
	<i>Total Unidade 3</i>		<i>8h</i>
<i>4 Semana</i>	<i>Unidade 4</i>	<i>Leitura do Módulo VI – Relacionamento (Parte2)</i>	<i>3h</i>
		<i>Aula no laboratório</i>	<i>1h</i>
		<i>Atividade 6 – Tarefa (XX exercícios)</i>	<i>3h</i>
		<i>Atividade 7 – Avaliação</i>	<i>1h</i>
	<i>Total Unidade 4</i>		<i>8h</i>
<i>5 Semana</i>	<i>Unidade 5</i>	<i>Leitura do Módulo VI – Relacionamento (Parte3)</i>	<i>3h</i>
		<i>Aula no laboratório</i>	<i>1h</i>
		<i>Atividade 8 – Tarefa (XX exercícios)</i>	<i>3h</i>
		<i>Atividade 9 – Avaliação</i>	<i>1h</i>
	<i>Total Unidade 5</i>		<i>8h</i>
	<i>Avaliação Final</i>	<i>Entrega das notas e feedback aos alunos</i>	<i>1h</i>
	<i>Avaliação do curso</i>	<i>Avaliação dos Alunos</i>	<i>1h</i>
	<i>Total Geral</i>		<i>40</i>
	<i>Agenda de Finalização</i>	<i>Encerramento do curso</i>	

Anexo F: Métricas - Análises de Correlação - 1º Ciclo

	NT	NC	NR	NRUna	NRBin	NRTer	NR:1N	NR:MN	NREsp	NRFor	NRFra	NCPk	NUK	NFK	NRNAry	NCNum	NCCrc	NCDtt	NCMsc	NCOpc	NCRqr	NRCrdPk11	NRCrdPk01	NRCrdPk1N	NRCrdPk0N	NRCrdFk11	NRCrdFk01	NRCrdFk1N	NRCrdFk0N	Nota
NT	1,00																													
NC	0,73	1																												
NR	0,80	0,71	1																											
NRUna	0,05	0,09	0,03	1																										
NRBin	0,56	0,51	0,63	0,02	1																									
NRTer	0,14	0,29	0,27	-0,02	0,25	1																								
NR:1N	0,29	0,39	0,35	0,07	0,2	0,17	1																							
NR:MN	0,26	0,41	0,27	0,08	0,38	0,29	0,46	1																						
NREsp	-0,02	0,18	0,12	0,08	0,16	0,32	0,22	0,33	1																					
NRFor	0,21	0,27	0,27	0,12	0,24	0,22	0,17	0,31	0,24	1																				
NRFra	0,42	0,49	0,5	0,06	0,48	0,29	0,24	0,38	0,2	0,5	1																			
NCPk	0,74	0,63	0,61	0,2	0,46	0,16	0,36	0,41	0,16	0,52	0,49	1																		
NUK	0,07	0,02	0,04	0,34	0,06	-0,02	0,03	0,01	-0	0,08	0,13	0,1	1																	
NFK	0,80	0,71	1	0,03	0,63	0,27	0,35	0,27	0,12	0,27	0,5	0,61	0,04	1																
NRNAry	-0,02	-0	0,08	-0,01	0,02	-0,03	-0,01	-0	-0,05	0,06	-0,03	-0,02	0,08	0,04	1															
NCNum	0,54	0,47	0,46	0,04	0,39	0,11	0,28	0,26	0,01	0,06	0,21	0,45	0,02	0,46	0,04	1														
NCCrc	0,29	0,34	0,36	-0,04	0,21	0,16	0,3	0,23	0,13	0,07	0,21	0,27	0,03	0,36	0,1	0,65	1													
NCDtt	0,13	0,36	0,26	0,06	0,14	0,33	0,26	0,32	0,43	0,32	0,32	0,28	-0,01	0,26	0,14	0,16	0,33	1												
NCMsc	0,05	-0	0,06	-0	-0,02	-0,01	0,03	-0,03	-0,02	0,04	-0,02	0,05	-0	0,06	-0	0,02	0,03	0,02	1											
NCOpc	-0,01	-0,02	-0	0,01	0,02	0,07	-0,02	0,02	0,21	0,07	0,08	0,03	-0,02	-0	-0,04	-0,12	-0,1	0,06	-0,01	1										
NCRqr	0,04	0,16	0,13	-0,04	0,02	0,22	-0	0,05	0,18	0,18	0,18	0,01	-0,08	0,13	0,05	-0,14	-0,04	0,34	0,02	0,42	1									
NRCrdPk11	0,31	0,41	0,4	0,06	0,33	0,25	0,51	0,51	0,35	0,22	0,27	0,36	0,01	0,4	0,01	0,29	0,29	0,34	0,03	-0,01	0,08	1								
NRCrdPk01	-0,17	-0,15	-0,21	-0,03	-0,18	0,03	-0,01	-0,25	-0,17	-0,17	-0,18	-0,24	-0,06	-0,21	-0,04	-0,1	-0,09	-0,15	-0,02	0,06	-0	-0,11	1							
NRCrdPk1N	-0,07	0,07	-0,06	0,06	0,03	0,26	0,22	0,29	0,21	0,1	0,12	0,07	-0,01	-0,06	0,05	0,09	0,05	0,18	-0,02	-0,08	0,05	0,2	1							
NRCrdPk0N	-0,04	0,03	-0,1	0,18	-0,07	0,15	0,14	0,31	0,29	0,14	0,1	0,13	0,04	-0,1	0,03	-0,02	-0	0,22	-0,01	0,16	0,19	0,1	-0	0,23	1					
NRCrdFk11	-0,02	0,01	-0,04	0,07	-0,01	0,13	0,08	0,2	0,24	0,14	0,07	0,1	0,02	-0,04	0,02	0,1	0,07	0,12	-0,02	0,01	0	0,08	-0,1	0,56	0,36	1				
NRCrdFk01	-0,08	0,14	0,04	0,11	0,1	0,29	0,24	0,35	0,8	0,21	0,19	0,11	0,02	0,04	-0	-0,03	0,07	0,4	-0,02	0,16	0,19	0,25	-0,12	0,28	0,39	0,25	1			
NRCrdFk1N	0,03	0,16	0,11	0,05	0,12	0,2	0,15	0,31	0,26	0,11	0,13	0,07	-0,09	0,11	0,01	0,04	0,03	0,18	-0,04	-0,01	0,09	0,39	-0,19	0,12	0,06	0,07	0,21	1		
NRCrdFk0N	0,16	0,06	0,13	-0,02	0,11	-0,01	0,18	0,05	0,11	0,04	0,13	0,14	-0,03	0,13	-0,04	0,03	-0,01	0,03	0,02	0,18	0,04	-0	0,29	-0,09	0,12	-0,08	0,11	0,13	1	
Nota	-0,36	-0,44	-0,44	-0,08	-0,42	-0,32	-0,6	-0,5	-0,46	-0,36	-0,47	-0,5	-0,03	-0,44	-0,07	-0,45	-0,4	-0,46	0,03	-0,06	-0,05	-0,48	0,1	-0,24	-0,25	-0,18	-0,41	-0,18	-0,17	1

Anexo G: Métricas - Análises de Correlação - 2º Ciclo

	NT	NC	NR	NRUna	NRBin	NRTer	NR:1N	NR:MN	NREsp	NRFor	NRFra	NCPk	NUK	NFK	NRNary	NCNum	NCCrc	NCDtt	NCMsc	NCOpc	NCRqr	NRCrdPk11	NRCrdPk01	NRCrdPk1N	NRCrdPk0N	NRCrdFk11	NRCrdFk01	NRCrdFk1N	NRCrdFk0N	Nota
NT	1																													
NC	0,66	1																												
NR	0,63	0,69	1																											
NRUna	0,22	0,07	0,04	1																										
NRBin	0,38	0,34	0,43	-0,02	1																									
NRTer	0,08	0,04	0,03	-0,04	0,21	1																								
NR:1N	0,35	0,4	0,48	-0,01	0,29	0,09	1																							
NR:MN	0,26	0,3	0,3	-0,01	0,46	0,1	0,51	1																						
NREsp	0,08	0,04	0,14	0,15	0,06	0,07	0,14	0,07	1																					
NRFor	0,29	0,29	0,34	0,05	0,28	0,01	0,3	0,31	0,1	1																				
NRFra	0,36	0,41	0,54	0,06	0,26	0	0,31	0,2	0,05	0,51	1																			
NCPk	0,71	0,57	0,56	0,14	0,35	0,1	0,38	0,31	0,1	0,58	0,49	1																		
NUK	0,01	-0,01	0,02	-0,01	0,01	-0,01	0,02	-0,02	0,01	0,01	0,04	0,02	1																	
NFK	0,63	0,69	1	0,04	0,43	0,03	0,48	0,3	0,14	0,34	0,54	0,56	0,02	1																
NRNary	0,04	0,12	0,23	-0	0,17	0,07	0,16	0,13	0,11	0,04	0,16	0,04	-0,01	0,23	1															
NCNum	0,52	0,55	0,51	0,01	0,25	0,03	0,34	0,28	0,01	0,27	0,28	0,44	-0	0,51	0,05	1														
NCCrc	0,37	0,56	0,37	0,04	0,17	0,01	0,22	0,19	-0,05	0,21	0,25	0,32	-0,01	0,37	0,03	0,49	1													
NCDtt	0,26	0,4	0,28	-0,02	0,11	0,11	0,16	0,12	0,09	0,22	0,19	0,27	0	0,28	0,05	0,24	0,28	1												
NCMsc	-0,02	-0,01	-0,02	-0,01	-0,01	0,01	-0,03	-0,03	-0,02	-0,02	-0,02	-0,03	-0	-0,02	-0,01	-0,02	-0,01	-0,01	0,01	1										
NCOpc	-0,16	-0,11	-0,07	-0,07	-0,02	0,05	-0,04	-0,05	0,05	-0,05	-0,03	-0,07	-0,02	-0,07	-0,02	-0,1	-0,06	0	-0,01	1										
NCRqr	0,2	0,33	0,24	-0,03	0,07	-0,05	0,16	0,03	-0,01	0,08	0,19	0,2	0,01	0,24	0	0,17	0,17	0,16	-0,02	0,27	1									
NRCrdPk11	0,35	0,37	0,46	0,12	0,26	0,11	0,71	0,39	0,21	0,26	0,27	0,35	0,03	0,46	0,14	0,29	0,22	0,15	-0,04	-0,05	0,14	1								
NRCrdPk01	-0,04	-0,03	-0,02	0,05	-0,03	0,08	0,02	-0,05	0,2	-0	-0,05	-0,02	-0	-0,02	0,17	-0,06	-0,04	0,03	-0,02	0,06	0,07	0,23	1							
NRCrdPk1N	0,03	-0,03	-0,06	0,01	0,02	0,08	0,36	0,23	0,02	0,05	0,01	0,03	0,03	-0,06	-0	0,03	0,01	-0,06	-0,02	-0,01	-0,07	0,33	-0,06	1						
NRCrdPk0N	0,08	-0,01	0,02	0,05	0,04	0,32	0,11	0,11	0,16	0,05	0,01	0,06	0	0,02	-0,04	-0	0,01	0,02	-0,01	0,08	-0,05	0,15	0,14	0,09	1					
NRCrdFk11	0,09	-0,03	-0,02	0,08	0,02	0,2	0,26	0,2	0,21	0,07	-0,01	0,07	-0	-0,02	0,04	0,06	-0,01	-0,02	-0,03	0,05	-0,1	0,2	0,06	0,55	0,36	1				
NRCrdFk01	0,01	-0,02	0,06	0,16	0	0,13	0,12	-0,01	0,64	0	-0,01	0,03	0,03	0,06	0,06	-0,06	-0,08	0,03	-0,02	0,06	0	0,19	0,22	0,08	0,27	0,19	1			
NRCrdFk1N	0,21	0,22	0,26	-0,04	0,14	0,15	0,51	0,32	0,17	0,21	0,2	0,25	-0	0,26	0,11	0,21	0,11	0,08	-0,04	0,06	0,18	0,48	0,07	0,19	0,16	0,25	0,16	1		
NRCrdFk0N	0,05	0,05	0,12	-0	0,05	0,05	0,15	0,1	0,24	0,1	0,08	0,1	0,03	0,12	0,07	0,03	-0,07	0,07	-0,03	0,11	0,21	0,09	0,26	-0,04	0,19	0,07	0,3	0,38	1	
Nota	-0,47	-0,47	-0,51	-0,13	-0,4	-0,17	-0,59	-0,49	-0,33	-0,47	-0,5	-0,53	-0,01	-0,51	-0,13	-0,46	-0,4	-0,21	0,04	0,03	-0,17	-0,59	-0,11	-0,26	-0,22	-0,31	-0,24	-0,47	-0,25	1

Anexo H: Resultado dos ensaios - 1º Ciclo

Tabela H-1. Resultados dos ensaios do primeiro ciclo (E = Ensaio, SD= desvio padrão).

PE	Algoritmo	Precision (%)	SD	Recall (%)	SD	F1-Score (%)	SD	Accuracy (%)	SD
E-1	ABC	64,800	5,497	61,818	5,838	62,347	5,545	61,818	5,838
E-2	ABC	65,178	1,982	63,498	3,156	63,573	2,598	63,498	3,156
E-3	ABC	64,107	2,983	62,411	3,018	62,523	2,331	62,411	3,018
E-4	ABC	63,552	3,431	61,806	5,174	61,791	4,336	61,806	5,174
E-5	ABC	61,852	2,347	56,849	6,966	57,695	5,462	56,849	6,966
E-6	ABC	59,737	3,462	49,504	12,613	51,006	11,401	49,504	12,613
E-7	ABC	59,881	4,316	55,315	9,451	55,799	7,975	55,315	9,451
E-8	ABC	57,239	5,608	51,543	10,110	52,565	9,357	51,543	10,110
E-9	ABC	49,569	8,981	49,398	10,088	47,027	9,683	49,398	10,088
E-1	DTC	78,802	4,698	78,106	4,941	77,956	4,763	78,106	4,941
E-2	DTC	77,699	2,002	77,643	1,745	77,451	1,894	77,643	1,745
E-3	DTC	76,767	1,467	76,599	1,554	76,545	1,435	76,599	1,554
E-4	DTC	75,880	1,981	75,760	1,891	75,645	1,973	75,760	1,891
E-5	DTC	74,018	1,327	73,744	1,462	73,759	1,333	73,744	1,462
E-6	DTC	72,867	1,210	72,614	1,138	72,422	1,138	72,614	1,138
E-7	DTC	72,864	1,232	72,587	1,155	72,514	1,134	72,587	1,155
E-8	DTC	68,838	1,818	68,164	1,963	68,272	1,923	68,164	1,963
E-9	DTC	65,292	2,991	63,401	2,725	63,850	2,685	63,401	2,725
E-1	GNB	58,547	1,485	44,470	2,744	44,614	2,540	44,470	2,744
E-2	GNB	59,380	1,891	44,619	2,058	45,120	2,525	44,619	2,058
E-3	GNB	59,380	1,891	44,619	2,058	45,120	2,525	44,619	2,058
E-4	GNB	59,019	2,060	46,027	2,866	46,130	2,826	46,027	2,866
E-5	GNB	57,475	1,276	44,399	2,946	44,612	2,693	44,399	2,946
E-6	GNB	58,762	2,404	44,124	2,715	44,781	2,737	44,124	2,715
E-7	GNB	60,088	1,731	46,500	4,265	47,591	4,101	46,500	4,265
E-8	GNB	62,207	2,751	45,928	5,365	47,838	5,737	45,928	5,365
E-9	GNB	60,567	1,671	46,117	4,245	48,229	4,634	46,117	4,245
E-1	K-NN	71,231	6,674	70,985	4,241	68,185	4,356	70,985	4,241
E-2	K-NN	69,646	4,103	69,810	3,518	67,723	3,431	69,810	3,518
E-3	K-NN	69,484	1,980	69,112	1,861	66,969	2,022	69,112	1,861
E-4	K-NN	68,296	1,668	68,517	1,242	66,027	1,384	68,517	1,242
E-5	K-NN	66,094	1,560	66,043	1,455	64,175	1,516	66,043	1,455
E-6	K-NN	65,162	2,633	65,292	2,446	63,320	2,575	65,292	2,446
E-7	K-NN	60,749	1,746	62,967	1,722	60,424	1,859	62,967	1,722
E-8	K-NN	59,133	3,669	60,666	1,393	56,621	1,510	60,666	1,393
E-9	K-NN	50,616	3,316	56,184	1,612	50,534	1,705	56,184	1,612
E-1	LDA	68,950	3,724	69,545	4,006	68,707	3,786	69,545	4,006
E-2	LDA	68,604	3,469	68,821	3,764	68,406	3,585	68,821	3,764
E-3	LDA	67,211	1,716	67,510	1,766	66,972	1,539	67,510	1,766
E-4	LDA	67,211	1,716	67,510	1,766	66,972	1,539	67,510	1,766
E-5	LDA	66,778	1,216	67,017	1,580	66,565	1,380	67,017	1,580
E-6	LDA	65,634	0,968	66,206	1,318	65,430	1,120	66,206	1,318
E-7	LDA	67,069	1,326	66,859	1,546	66,681	1,423	66,859	1,546
E-8	LDA	65,313	2,047	64,567	2,292	64,630	2,098	64,567	2,292
E-9	LDA	62,675	2,586	61,235	2,989	61,614	2,752	61,235	2,989

Tabela H-2. Resultados dos ensaios do primeiro ciclo (E = Ensaio, SD= desvio padrão).

E	Algoritmo	Precision (%)	SD	Recall (%)	SD	F1-Score (%)	SD	Accuracy (%)	SD
PE-1	LR	70,888	5,148	64,470	6,282	65,527	5,720	64,470	6,282
PE-2	LR	70,420	2,290	64,601	3,346	65,867	2,764	64,601	3,346
PE-3	LR	69,122	2,153	64,264	2,176	65,194	2,070	64,264	2,176
PE-4	LR	69,074	1,375	63,403	1,701	64,799	1,450	63,403	1,701
PE-5	LR	69,194	1,163	63,896	0,965	64,959	1,040	63,896	0,965
PE-6	LR	67,630	1,059	62,297	1,637	63,531	1,630	62,297	1,637
PE-7	LR	67,298	2,376	62,989	2,562	64,070	2,338	62,989	2,562
PE-8	LR	65,948	1,597	61,180	1,791	62,564	1,846	61,180	1,791
PE-9	LR	64,479	2,612	60,592	2,753	61,618	2,668	60,592	2,753
PE-1	MLP	80,911	4,156	80,455	4,253	80,217	4,137	80,455	4,253
PE-2	MLP	79,128	2,210	79,354	1,964	78,962	2,106	79,354	1,964
PE-3	MLP	78,487	1,650	78,731	1,591	78,431	1,550	78,731	1,591
PE-4	MLP	78,027	1,524	78,308	1,485	77,973	1,512	78,308	1,485
PE-5	MLP	76,304	1,920	76,667	1,909	76,255	1,900	76,667	1,909
PE-6	MLP	74,957	1,510	75,190	1,754	74,761	1,588	75,190	1,754
PE-7	MLP	75,140	1,164	75,402	1,304	75,014	1,138	75,402	1,304
PE-8	MLP	71,409	1,570	71,656	1,645	71,254	1,629	71,656	1,645
PE-9	MLP	66,862	2,056	67,166	2,329	66,656	2,112	67,166	2,329
PE-1	RFC	81,485	4,426	81,061	4,374	80,577	4,088	81,061	4,374
PE-2	RFC	80,300	2,642	80,563	2,232	79,992	2,546	80,563	2,232
PE-3	RFC	79,234	1,619	79,721	1,506	79,071	1,519	79,721	1,506
PE-4	RFC	79,725	1,371	80,038	1,347	79,394	1,343	80,038	1,347
PE-5	RFC	78,258	1,283	78,752	1,339	78,116	1,289	78,752	1,339
PE-6	RFC	76,814	1,719	77,386	1,750	76,460	1,712	77,386	1,750
PE-7	RFC	76,420	1,197	77,163	1,038	76,199	1,040	77,163	1,038
PE-8	RFC	73,854	1,695	74,843	1,619	73,739	1,723	74,843	1,619
PE-9	RFC	69,173	2,037	70,584	1,599	69,109	1,731	70,584	1,599
PE-1	SGD	61,270	7,639	61,045	5,534	57,846	5,806	61,045	5,534
PE-2	SGD	62,319	5,603	61,622	4,875	58,696	4,246	61,622	4,875
PE-3	SGD	62,747	5,105	60,731	5,114	58,083	4,929	60,731	5,114
PE-4	SGD	61,948	5,343	60,194	5,418	57,223	4,509	60,194	5,418
PE-5	SGD	61,485	5,271	59,925	5,344	57,257	4,656	59,925	5,344
PE-6	SGD	61,990	5,031	60,633	4,814	57,822	4,377	60,633	4,814
PE-7	SGD	61,919	4,639	59,809	4,697	57,356	4,204	59,809	4,697
PE-8	SGD	61,344	4,910	58,961	6,315	56,451	5,577	58,961	6,315
PE-9	SGD	59,311	4,668	57,508	4,863	54,833	4,060	57,508	4,863
PE-1	SVC	80,040	3,995	80,076	3,864	78,750	3,944	80,076	3,864
PE-2	SVC	79,183	2,130	79,696	1,957	78,440	2,363	79,696	1,957
PE-3	SVC	78,178	1,629	78,629	1,493	77,374	1,574	78,629	1,493
PE-4	SVC	77,051	1,819	77,586	1,471	75,989	1,596	77,586	1,471
PE-5	SVC	76,616	1,391	76,865	1,263	75,110	1,490	76,865	1,263
PE-6	SVC	74,981	2,018	75,343	1,542	73,188	1,646	75,343	1,542
PE-7	SVC	74,383	0,971	74,228	0,843	71,712	1,127	74,228	0,843
PE-8	SVC	71,126	3,071	70,961	2,148	67,694	2,457	70,961	2,148
PE-9	SVC	67,434	3,255	66,819	1,152	62,154	1,676	66,819	1,152

Anexo I: Resultado dos ensaios - 2º Ciclo

Tabela I-1. Resultados dos ensaios do segundo ciclo (E = Ensaio, SD= desvio padrão).

E	Algoritmo	Precision (%)	SD	Recall (%)	SD	F1-Score (%)	SD	Accuracy (%)	SD
E-1	ABC	46,036	13,311	37,945	10,659	33,978	10,987	37,945	10,659
E-2	ABC	41,038	5,596	34,380	1,714	30,542	1,583	34,380	1,714
E-3	ABC	46,447	13,361	41,146	13,391	37,361	13,726	41,146	13,391
E-4	ABC	51,019	11,373	44,759	16,070	41,758	16,411	44,759	16,070
E-5	ABC	49,890	11,736	45,666	15,617	42,435	16,569	45,666	15,617
E-6	ABC	51,089	12,946	44,667	16,044	41,830	16,766	44,667	16,044
E-7	ABC	56,170	10,086	48,289	16,805	45,481	17,749	48,289	16,805
E-8	ABC	55,960	13,066	46,772	18,097	44,288	18,444	46,772	18,097
E-9	ABC	59,058	7,475	52,526	14,429	51,177	13,885	52,526	14,429
E-1	DTC	92,271	0,911	92,119	0,957	92,131	0,942	92,119	0,957
E-2	DTC	91,685	0,649	91,633	0,651	91,637	0,647	91,633	0,651
E-3	DTC	91,058	0,620	90,884	0,660	90,931	0,632	90,884	0,660
E-4	DTC	90,826	0,674	90,653	0,670	90,709	0,666	90,653	0,670
E-5	DTC	89,932	0,619	89,737	0,666	89,797	0,641	89,737	0,666
E-6	DTC	88,723	0,608	88,398	0,684	88,503	0,648	88,398	0,684
E-7	DTC	87,499	0,849	87,399	0,918	87,406	0,896	87,399	0,918
E-8	DTC	85,149	0,665	84,711	0,823	84,865	0,757	84,711	0,823
E-9	DTC	80,771	1,124	80,007	1,327	80,258	1,232	80,007	1,327
E-1	GNB	74,463	1,232	69,492	1,570	70,717	1,461	69,492	1,570
E-2	GNB	74,205	0,896	68,975	1,187	70,255	1,143	68,975	1,187
E-3	GNB	74,205	0,896	68,975	1,187	70,255	1,143	68,975	1,187
E-4	GNB	74,022	0,980	68,881	1,061	70,152	1,057	68,881	1,061
E-5	GNB	73,773	0,936	68,625	0,785	69,904	0,860	68,625	0,785
E-6	GNB	73,837	1,177	68,777	1,093	69,998	1,204	68,777	1,093
E-7	GNB	73,614	0,740	68,836	0,687	70,018	0,730	68,836	0,687
E-8	GNB	72,768	1,795	66,655	2,054	67,638	2,270	66,655	2,054
E-9	GNB	71,935	2,309	65,610	1,841	66,476	2,076	65,610	1,841
E-1	K-NN	79,721	1,683	78,453	1,680	78,646	1,716	78,453	1,680
E-2	K-NN	78,194	1,125	76,649	1,197	76,895	1,180	76,649	1,197
E-3	K-NN	78,282	0,738	76,959	0,641	77,161	0,684	76,959	0,641
E-4	K-NN	77,284	1,233	76,090	1,047	76,198	1,059	76,090	1,047
E-5	K-NN	76,624	1,298	75,870	1,040	75,731	1,167	75,870	1,040
E-6	K-NN	74,137	1,488	73,868	0,956	73,445	1,204	73,868	0,956
E-7	K-NN	71,770	1,204	72,019	0,763	71,197	0,892	72,019	0,763
E-8	K-NN	68,322	1,891	69,454	1,189	68,048	1,389	69,454	1,189
E-9	K-NN	60,293	2,222	63,242	1,840	59,993	1,919	63,242	1,840
E-1	LDA	74,860	1,444	74,237	1,191	74,145	1,300	74,237	1,191
E-2	LDA	75,027	1,335	74,507	1,392	74,383	1,389	74,507	1,392
E-3	LDA	75,016	1,033	74,462	1,058	74,345	1,061	74,462	1,058
E-4	LDA	75,016	1,033	74,462	1,058	74,345	1,061	74,462	1,058
E-5	LDA	75,032	1,429	74,588	1,464	74,403	1,503	74,588	1,464
E-6	LDA	74,513	1,436	74,077	1,437	73,898	1,475	74,077	1,437
E-7	LDA	74,176	1,084	73,816	1,092	73,587	1,100	73,816	1,092
E-8	LDA	74,042	1,923	73,528	1,628	73,359	1,781	73,528	1,628
E-9	LDA	72,218	1,712	71,717	1,029	71,493	1,346	71,717	1,029

Tabela I-2. Resultados dos ensaios do segundo ciclo (E = Ensaio, SD= desvio padrão).

E	Algoritmo	Precision (%)	SD	Recall (%)	SD	F1-Score (%)	SD	Accuracy (%)	SD
E-1	LR	81,947	1,436	78,157	1,285	79,100	1,352	78,157	1,285
E-2	LR	81,388	1,048	77,423	1,356	78,449	1,239	77,423	1,356
E-3	LR	81,833	1,040	77,546	1,164	78,564	1,107	77,546	1,164
E-4	LR	81,153	1,016	77,268	1,031	78,211	1,035	77,268	1,031
E-5	LR	81,227	0,759	77,339	0,949	78,312	0,893	77,339	0,949
E-6	LR	80,514	1,222	76,605	1,378	77,534	1,348	76,605	1,378
E-7	LR	79,631	1,192	75,820	1,107	76,690	1,105	75,820	1,107
E-8	LR	79,374	0,904	75,653	0,914	76,547	0,854	75,653	0,914
E-9	LR	76,148	0,954	72,674	1,244	73,467	1,246	72,674	1,244
E-1	MLP	92,097	0,900	92,055	0,889	92,016	0,879	92,055	0,889
E-2	MLP	91,591	0,730	91,559	0,693	91,528	0,712	91,559	0,693
E-3	MLP	91,506	0,655	91,471	0,681	91,457	0,670	91,471	0,681
E-4	MLP	90,786	0,673	90,753	0,698	90,754	0,678	90,753	0,698
E-5	MLP	90,919	0,624	90,883	0,597	90,881	0,606	90,883	0,597
E-6	MLP	89,780	0,611	89,653	0,647	89,675	0,626	89,653	0,647
E-7	MLP	88,783	0,511	88,733	0,509	88,722	0,503	88,733	0,509
E-8	MLP	87,100	0,453	87,125	0,409	87,083	0,422	87,125	0,409
E-9	MLP	83,771	0,561	83,940	0,714	83,814	0,623	83,940	0,714
E-1	RFC	92,899	0,814	92,860	0,793	92,815	0,820	92,860	0,793
E-2	RFC	92,021	0,802	92,070	0,819	91,993	0,812	92,070	0,819
E-3	RFC	91,422	0,855	91,485	0,844	91,404	0,858	91,485	0,844
E-4	RFC	91,409	0,596	91,475	0,605	91,397	0,593	91,475	0,605
E-5	RFC	90,758	0,731	90,887	0,690	90,759	0,713	90,887	0,690
E-6	RFC	89,924	0,526	90,018	0,517	89,921	0,523	90,018	0,517
E-7	RFC	89,453	0,740	89,606	0,682	89,458	0,701	89,606	0,682
E-8	RFC	88,143	0,669	88,424	0,596	88,151	0,616	88,424	0,596
E-9	RFC	84,643	0,851	85,006	0,895	84,613	0,932	85,006	0,895
E-1	SGD	68,794	5,561	68,066	5,732	65,605	6,134	68,066	5,732
E-2	SGD	68,775	5,858	67,917	5,575	64,947	6,355	67,917	5,575
E-3	SGD	69,178	5,657	68,100	5,174	65,538	6,212	68,100	5,174
E-4	SGD	69,149	4,628	68,197	4,719	65,581	5,182	68,197	4,719
E-5	SGD	68,009	5,053	67,488	5,210	64,731	5,695	67,488	5,210
E-6	SGD	67,091	5,884	66,225	5,594	63,001	6,009	66,225	5,594
E-7	SGD	67,154	5,750	67,485	4,466	64,451	5,673	67,485	4,466
E-8	SGD	67,285	5,249	66,959	5,241	64,000	5,794	66,959	5,241
E-9	SGD	64,530	6,154	64,728	5,815	61,124	6,508	64,728	5,815
E-1	SVC	88,618	1,513	88,792	1,382	88,502	1,451	88,792	1,382
E-2	SVC	87,636	0,751	88,017	0,671	87,600	0,774	88,017	0,671
E-3	SVC	87,575	0,647	87,935	0,624	87,568	0,660	87,935	0,624
E-4	SVC	87,137	0,573	87,512	0,508	87,084	0,512	87,512	0,508
E-5	SVC	86,566	0,984	87,016	0,878	86,468	0,919	87,016	0,878
E-6	SVC	85,215	0,732	85,732	0,631	85,056	0,611	85,732	0,631
E-7	SVC	83,722	0,616	84,323	0,574	83,468	0,697	84,323	0,574
E-8	SVC	81,142	1,130	81,971	1,091	80,531	1,297	81,971	1,091
E-9	SVC	76,567	1,661	77,840	1,392	75,564	1,623	77,840	1,392

Anexo J: Resultado dos ensaios - 3º Ciclo

Tabela J-1. Resultados dos ensaios do terceiro ciclo (E = Ensaio, SD= desvio padrão).

E	Algoritmo	Precision (%)	SD	Recall (%)	SD	F1-Score (%)	SD	Accuracy (%)	SD
E-1	ABC	67,700	2,702	68,347	3,512	67,025	3,211	68,347	3,512
E-2	ABC	69,324	3,404	68,813	2,578	67,928	2,964	68,813	2,578
E-3	ABC	71,077	5,067	72,037	2,810	70,158	3,912	72,037	2,810
E-4	ABC	68,323	3,390	67,369	3,388	66,055	1,983	67,369	3,388
E-5	ABC	69,524	3,577	68,455	2,042	67,451	2,393	68,455	2,042
E-6	ABC	68,325	5,249	69,346	2,663	67,735	4,323	69,346	2,663
E-7	ABC	69,758	4,275	69,370	2,549	68,276	3,488	69,370	2,549
E-8	ABC	66,115	6,363	66,606	2,857	65,085	5,016	66,606	2,857
E-9	ABC	60,177	9,443	52,354	12,155	51,336	12,718	52,354	12,155
E-1	DTC	92,865	1,498	92,669	1,623	92,723	1,569	92,669	1,623
E-2	DTC	92,744	0,602	92,630	0,559	92,662	0,575	92,630	0,559
E-3	DTC	92,055	0,822	91,860	0,911	91,918	0,882	91,860	0,911
E-4	DTC	91,601	0,462	91,459	0,473	91,500	0,464	91,459	0,473
E-5	DTC	90,765	0,510	90,552	0,562	90,622	0,540	90,552	0,562
E-6	DTC	89,733	0,429	89,519	0,389	89,594	0,394	89,519	0,389
E-7	DTC	88,465	1,159	88,190	1,238	88,287	1,212	88,190	1,238
E-8	DTC	86,339	0,919	85,984	0,971	86,096	0,935	85,984	0,971
E-9	DTC	81,953	1,204	81,342	1,261	81,544	1,218	81,342	1,261
E-1	GNB	71,352	1,923	64,767	1,871	65,699	1,860	64,767	1,871
E-2	GNB	70,787	1,230	63,784	1,656	64,836	1,552	63,784	1,656
E-3	GNB	70,787	1,230	63,784	1,656	64,836	1,552	63,784	1,656
E-4	GNB	70,942	0,853	63,905	1,281	64,970	1,100	63,905	1,281
E-5	GNB	71,115	1,245	63,773	1,153	65,061	1,169	63,773	1,153
E-6	GNB	70,594	1,506	63,550	1,639	64,601	1,730	63,550	1,639
E-7	GNB	70,232	1,510	63,559	1,613	64,638	1,666	63,559	1,613
E-8	GNB	70,269	1,715	61,764	1,210	62,979	1,414	61,764	1,210
E-9	GNB	70,280	1,905	61,066	2,324	62,460	2,149	61,066	2,324
E-1	K-NN	82,980	0,585	81,992	0,655	82,178	0,595	81,992	0,655
E-2	K-NN	82,188	0,886	81,145	0,852	81,353	0,862	81,145	0,852
E-3	K-NN	82,189	0,939	80,891	1,118	81,207	1,071	80,891	1,118
E-4	K-NN	81,040	0,562	79,528	0,724	79,855	0,647	79,528	0,724
E-5	K-NN	79,520	0,665	78,162	0,671	78,326	0,560	78,162	0,671
E-6	K-NN	76,893	1,151	75,884	1,083	75,710	1,004	75,884	1,083
E-7	K-NN	75,081	0,789	74,207	0,939	73,919	0,918	74,207	0,939
E-8	K-NN	72,708	1,847	72,653	1,059	72,098	1,454	72,653	1,059
E-9	K-NN	65,653	1,277	67,708	1,214	65,444	1,054	67,708	1,214
E-1	LDA	79,690	1,023	78,517	1,144	78,823	1,096	78,517	1,144
E-2	LDA	78,706	1,215	77,699	1,334	77,966	1,284	77,699	1,334
E-3	LDA	78,651	0,855	77,687	0,945	77,919	0,924	77,687	0,945
E-4	LDA	78,651	0,855	77,687	0,945	77,919	0,924	77,687	0,945
E-5	LDA	78,724	0,760	77,772	0,844	78,008	0,814	77,772	0,844
E-6	LDA	78,296	1,116	77,362	1,151	77,555	1,189	77,362	1,151
E-7	LDA	78,155	0,883	77,329	0,734	77,478	0,818	77,329	0,734
E-8	LDA	77,192	2,023	76,501	1,901	76,572	2,011	76,501	1,901
E-9	LDA	76,240	1,461	75,093	1,757	75,283	1,645	75,093	1,757

Tabela J-2. Resultados dos ensaios do terceiro ciclo (E = Ensaio, SD= desvio padrão).

E	Algoritmo	Precision (%)	SD	Recall (%)	SD	F1-Score (%)	SD	Accuracy (%)	SD
E-1	LR	84,198	0,842	80,826	0,825	81,736	0,729	80,826	0,825
E-2	LR	84,138	0,965	80,859	0,868	81,752	0,854	80,859	0,868
E-3	LR	84,438	0,688	80,863	0,949	81,799	0,850	80,863	0,949
E-4	LR	83,029	1,378	79,560	1,388	80,471	1,399	79,560	1,388
E-5	LR	82,774	1,443	79,372	1,813	80,262	1,772	79,372	1,813
E-6	LR	82,571	1,358	79,070	1,512	79,952	1,509	79,070	1,512
E-7	LR	81,612	1,659	78,303	1,534	79,142	1,582	78,303	1,534
E-8	LR	80,774	1,017	77,284	0,741	78,150	0,759	77,284	0,741
E-9	LR	78,506	0,850	75,055	1,118	75,965	1,014	75,055	1,118
E-1	MLP	93,020	0,835	92,987	0,802	92,972	0,822	92,987	0,802
E-2	MLP	92,655	0,463	92,683	0,441	92,645	0,464	92,683	0,441
E-3	MLP	92,404	0,808	92,440	0,784	92,400	0,795	92,440	0,784
E-4	MLP	91,901	0,519	91,931	0,553	91,889	0,536	91,931	0,553
E-5	MLP	91,908	0,623	91,948	0,617	91,907	0,614	91,948	0,617
E-6	MLP	91,230	0,342	91,206	0,371	91,203	0,346	91,206	0,371
E-7	MLP	89,104	0,743	89,124	0,613	89,083	0,661	89,124	0,613
E-8	MLP	88,767	0,467	88,793	0,460	88,741	0,444	88,793	0,460
E-9	MLP	85,774	0,733	85,763	0,612	85,720	0,660	85,763	0,612
E-1	RFC	93,304	0,686	93,305	0,671	93,230	0,702	93,305	0,671
E-2	RFC	92,692	0,917	92,750	0,888	92,677	0,917	92,750	0,888
E-3	RFC	92,529	0,733	92,603	0,728	92,513	0,737	92,603	0,728
E-4	RFC	92,375	0,670	92,440	0,657	92,344	0,666	92,440	0,657
E-5	RFC	91,833	0,504	91,952	0,517	91,815	0,498	91,952	0,517
E-6	RFC	90,943	0,649	91,040	0,605	90,921	0,612	91,040	0,605
E-7	RFC	90,248	0,432	90,412	0,439	90,225	0,459	90,412	0,439
E-8	RFC	88,935	0,702	89,183	0,630	88,939	0,673	89,183	0,630
E-9	RFC	85,790	0,678	86,072	0,747	85,727	0,744	86,072	0,747
E-1	SGD	69,898	6,042	69,208	4,658	66,148	5,360	69,208	4,658
E-2	SGD	69,898	6,778	68,191	5,631	65,338	6,397	68,191	5,631
E-3	SGD	69,233	6,469	68,820	5,171	65,679	6,073	68,820	5,171
E-4	SGD	69,662	5,524	68,249	5,641	65,358	6,240	68,249	5,641
E-5	SGD	70,221	5,041	68,343	4,777	65,385	5,484	68,343	4,777
E-6	SGD	68,979	6,118	68,151	4,550	65,123	5,652	68,151	4,550
E-7	SGD	68,270	6,085	68,117	5,024	64,725	5,771	68,117	5,024
E-8	SGD	67,117	6,653	66,501	5,626	62,672	6,690	66,501	5,626
E-9	SGD	63,272	8,346	63,784	6,793	59,562	7,632	63,784	6,793
E-1	SVC	89,668	1,271	89,873	1,172	89,682	1,248	89,873	1,172
E-2	SVC	89,860	0,750	90,127	0,729	89,900	0,741	90,127	0,729
E-3	SVC	89,546	0,808	89,802	0,747	89,597	0,790	89,802	0,747
E-4	SVC	89,052	0,645	89,342	0,607	89,101	0,632	89,342	0,607
E-5	SVC	88,708	0,787	89,058	0,734	88,765	0,732	89,058	0,734
E-6	SVC	87,791	0,572	88,165	0,537	87,840	0,534	88,165	0,537
E-7	SVC	86,969	0,517	87,426	0,432	87,035	0,531	87,426	0,432
E-8	SVC	85,387	0,809	86,048	0,717	85,336	0,802	86,048	0,717
E-9	SVC	81,646	1,214	82,558	1,156	81,633	1,182	82,558	1,156

Anexo K: Configurações para a concepção do modelo de aprendizagem

Tabela K-1. Configuração dos algoritmos de aprendizagem computacional scikit-learn.org.

Algoritmo	Configurações
MLP	RS=1, hls=(50), max_iter=2000,
LR	multi_class='multinomial', solver='newton-cg', max_iter=5, class_weight='balanced', C=1
LDA	Configuração padrão
K-NN	n_neighbors=30, weights='distance', leaf_size=5, algorithm='auto',
SGD	alpha=0.00001, n_iter=2, epsilon=0.25, loss='modified_huber'
RFC	n_estimators=100, max_depth=10, random_state = 0
DTC	random_state=0
GNB	Configuração padrão
SVC	gamma=2, C=1, probability=True
ABC	Configuração padrão
Lr-Learning Rate; iter-Iterations; comp-Number of Hidden Components of RBM; RS-Random State hls=Hidden Layer Sizes; CLS-Number of clusters at the input layer of RANs	