

Mestrado em Engenharia Informática

Dissertação/Estágio

Relatório Final

Concepção de Software para Gestão de Horários Escolares

Orientando:

Engenheiro Alexandre R. S. Correia

arsc@student.dei.uc.pt

Orientador:

Professor Doutor Filipe Araújo

filipius@dei.uc.pt

Data: 10 de Julho de 2009



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Índice

Índice.....	2
Lista de figuras.....	5
Abstract.....	7
Keywords.....	7
Resumo.....	8
Palavras-chave.....	8
Definições e Acrónimos.....	9
Agradecimentos.....	9
Capítulo 1. Introdução.....	10
1.1 Âmbito.....	10
1.2 Objectivos.....	10
1.3 Resultados.....	10
1.4 Organização do documento.....	11
Capítulo 2. Estado da arte.....	13
2.1 Sistemas de gestão de horários dos cursos.....	13
2.1.1 Elaborar o horário.....	13
2.1.2 Elaborar e comunicar o horário dos cursos.....	13
2.2 Apresentação de duas opções existentes.....	14
2.2.1 Parâmetros para avaliar as opções.....	14
2.2.2 UniTime.....	15
2.2.3 FET.....	16
2.3 Fraquezas das opções apresentadas.....	17
2.3.1 UniTime.....	17
2.3.2 FET.....	18
2.3.3 Aplicação self.....	19
2.4 Resultados da avaliação.....	20
2.5 Sistema actual do DEI.....	20
2.5.1 Caracterização.....	20
2.5.2 Arquitectura.....	20
2.5.3 Fraquezas.....	21
2.5.4 Módulos.....	22
2.5.4.1 Add subject.....	22
2.5.4.2 Delete subject.....	22
2.5.4.3 Add new name to subject.....	22
2.5.4.4 Delete name to subject.....	22
2.5.4.5 Add Course-Year.....	22
2.5.4.6 Delete Course-Year.....	22
2.5.4.7 Add & Delete Subject to Course-Year.....	23
2.5.4.8 Add Professors.....	23
2.5.4.9 Delete Professors.....	23
2.5.4.10 Add Professors Preferences.....	23

2.5.4.11 Delete Professors Preferences.....	24
2.5.4.12 Add Type of Class.....	24
2.5.4.13 Delete Type of Class.....	24
2.5.4.14 Add Class of Subject.....	24
2.5.4.15 Delete Class of Subject.....	24
2.5.4.16 Assign professors.....	25
2.5.4.17 Add Room.....	25
2.5.4.18 Delete Room.....	25
2.5.4.19 Add Schedule to Time Table.....	26
2.5.4.20 Delete Schedule to Time Table.....	27
2.5.4.21 See Professors Time Table.....	27
2.5.4.22 See Rooms Time Table.....	27
Capítulo 3. Proposta do projecto.....	28
3.1 Conceito.....	28
3.2 Termos.....	28
3.2.1 Objectos e recursos.....	29
3.2.1.1 Período (Period).....	29
3.2.1.2 Slot.....	29
3.2.1.3 Curso (Course).....	29
3.2.1.4 Disciplina (Subject).....	29
3.2.1.5 Tipo de aula (Typeclass).....	30
3.2.1.6 Grupo de utilizador (GroupUser).....	30
3.2.1.7 Utilizador (User).....	30
3.2.1.8 Sala (Room).....	30
3.2.1.9 Aula (Currentclass).....	30
3.2.1.10 Aula Utilizador (Currentclass_User).....	30
3.2.1.11 Slot Utilizador (Slot_User).....	31
3.2.1.12 Horário (Timetable).....	31
3.2.2 Conjunto de restrições.....	31
3.3 Funcionalidades (módulos).....	32
3.3.1 Gestão de módulos tipo.....	32
3.3.2 Gestão de Slots.....	32
3.3.3 Gestão de Horários.....	32
3.3.4 Geração de ficheiros para exportação.....	32
3.4 Arquitectura.....	32
Capítulo 4. Metodologia de abordagem.....	34
4.1 Desenvolvimento iterativo e incremental.....	34
4.2 Plano de trabalho.....	35
4.3 Artefactos para entregar ao cliente.....	36
4.4 Componentes.....	36
4.4.1 Servidor de base de dados.....	37
4.4.2 Servidor Web.....	37
4.4.3 Linguagem da aplicação.....	37
4.4.4 Framework PHP.....	37
4.4.5 Integrated Development Environment (IDE).....	37
4.4.6 Cliente da aplicação.....	37
Capítulo 5. Trabalho realizado.....	38
5.1 Requisitos.....	38
5.1.1 Funcionais.....	38

5.1.2 Não funcionais (qualidade).....	38
5.2 Design.....	38
5.2.1 Classes.....	39
5.2.1.1 Classes tipo.....	39
5.2.1.2 Outras classes.....	39
5.2.2 Actores.....	39
5.2.2.1 Equipa de gestão do horário.....	39
5.2.2.2 Professores.....	39
5.2.2.3 Público.....	39
5.2.2.4 Aplicação de utilização do DEI.....	40
5.2.3 Casos de uso.....	40
5.2.3.1 Login na área restrita.....	40
5.2.3.2 Operações em classes tipo.....	40
5.2.3.3 Operações em outras classes.....	41
5.2.4 Casos de uso de relatórios.....	41
5.2.4.1 Na área pública.....	41
5.2.4.2 Na área restrita.....	42
5.2.5 Base de dados.....	42
5.2.5.1 Estrutura.....	42
5.2.5.2 Integridade dos dados.....	43
Capítulo 6. Conclusões e perspectivas futuras.....	44
6.1 Caracterização.....	44
6.2 Resultados esperados.....	44
6.3 Lições aprendidas.....	44
6.4 Trabalho futuro.....	45
Capítulo 7. Referências.....	46
Capítulo 8. Anexos.....	49
8.1 Product Backlog.....	49
8.1.1 Requisitos funcionais.....	49
8.1.2 Requisitos não funcionais (qualidade).....	50
8.2 Especificação dos testes.....	50
8.2.1 Descrição da tabela de testes.....	50
8.2.2 Nomenclatura.....	51
8.2.3 Tabela de testes.....	51
8.2.3.1 Relatório List per Course (área pública).....	51
8.2.3.2 Login na área restrita.....	52
8.2.3.3 Pesquisar (search) no módulo de períodos (Periods).....	52
8.2.3.4 Adicionar um novo período (Periods).....	52
8.2.3.5 Editar o conteúdo de um período (Periods).....	53
8.2.3.6 Visualizar o conteúdo de um período (Periods).....	54
8.2.3.7 Excluir um ou mais períodos (Periods).....	54
8.2.3.8 Operações nos módulos Courses, Subjects, Typeclasses, Group_users, Users, Rooms e Slot_users.....	55
8.2.3.9 Adicionar uma nova aula (Currentclasses).....	55
8.2.3.10 Demais operações em aulas (Currentclasses).....	56
8.2.3.11 Cópia de slots entre períodos (Copy between periods).....	56
8.2.3.12 Demais operações em Slots.....	56
8.2.3.13 Adicionar novos horários (Timetables).....	57
8.2.3.14 Excluir horários (Timetables) existentes.....	57

8.2.3.15 Relatório LUOS (List User's Overlaps per Slot).....	58
8.2.3.16 Demais relatórios do módulo de horários (Timetables).....	58
8.3 Diagrama da base de dados.....	59
8.4 Definição da base de dados (script).....	60
8.5 Guia de início rápido do utilizador.....	69
8.5.1 Como aceder a aplicação.....	69
8.5.2 Como gerar os relatórios na área pública.....	69
8.5.3 Como ligar e desligar da área restrita.....	70
8.5.4 Como fazer pesquisa.....	70
8.5.4.1 Opções de pesquisa.....	71
8.5.5 Como adicionar um novo período.....	71
8.5.6 Como alterar o conteúdo de um período.....	72
8.5.7 Como visualizar o conteúdo de um período.....	72
8.5.8 Como excluir períodos.....	72
8.5.9 Operações SCRUD em outros módulos.....	73
8.5.10 Como adicionar uma aula (currentclass).....	73
8.5.11 Como adicionar um horário (timetable).....	74
8.5.12 Como excluir um horário (timetable).....	75

Lista de figuras

Ilustração 1 – Edição do recurso sala (Room) 7.....	18
Ilustração 2 – alguns parâmetros do FET timetable software.....	19
Ilustração 3 – Resultado da avaliação.....	20
Ilustração 4 – Curso ano.....	23
Ilustração 5 – Preferências do professor.....	24
Ilustração 6 – Professores das disciplinas.....	25
Ilustração 7 – Adicionar horário.....	26
Ilustração 8 – Adicionar sala no horário.....	26
Ilustração 9 – Exemplo de grelha com slots.....	29
Ilustração 10 – Arquitectura da aplicação.....	33
Ilustração 11 – Diagrama de Gantt do plano de trabalho.....	35
Ilustração 12 – Excerto da página inicial do sítio Web (Home).....	38
Ilustração 13 – Gestão da classe tipo “Período”.....	40
Ilustração 14 – Método adicionar da classe horário.....	41
Ilustração 15 – Relatório horário por curso.....	42
Ilustração 16 – Estrutura da base de dados (excerto).....	42
Ilustração 17 – Página inicial da aplicação Web.....	69
Ilustração 18 – Relatório de horários por curso.....	69
Ilustração 19 – Pesquisa (search) de períodos.....	70
Ilustração 20 – Adicionar novo período.....	72

Ilustração 21 – Alterar conteúdo de um período.....	72
Ilustração 22 – Visualizar conteúdo de um período.....	72
Ilustração 23 – Excluir períodos.....	73
Ilustração 24 – Adicionar uma aula.....	73
Ilustração 25 – Adicionar um horário.....	75
Ilustração 26 – Excluir um horário.....	75

Abstract

This project is inserted in the Dissertation / Stage academic year 2008/09 belonging to the Master of Engineering (MEI) of the Department of Informatics Engineering (DEI), Faculty of Science and Technology, University of Coimbra (FCTUC).

Saved some special features, the task of drawing up schedules of academic courses and then communicate to the whole DEI community hasn't been different from what occurs in other university departments around the world. Since the organization of the manual classes, disciplines, courses, teachers and classrooms, a weekly schedule of times, is something that consumes a significant amount of time, the benefits of adequate IT systems are evident.

Thus, the main objective of this project was to improve the performance of tasks to develop and communicate the timetable of courses offered by the Department. Based on one analysis of the problem and the current computer system, it was developed a new application, called TTSv2, which is ready for development and communication of schedules in the next academic year.

The author of this document will try to convince the reader that the application TTSv2 is an improvement on the current state, it includes enhancements such as: **Security** - communication between the Web server and the client browsers is secure as it using protocols SSL / TLS. **Users now have credentials to access** the application and carry out tasks; **Less manual intervention** - the amount of manual intervention that occurred previously been reduced, making the process more efficient; **available for use** - the application is available online for simultaneous use by the whole community DEI, and can be accessed from anywhere via the URL <https://timetable.dei.uc.pt/timetable/> , and; **maintenance and extensibility** - the project followed the "design patterns" model view controller (MVC) and recommendations of the Framework adopted. Thus it is believed that with less effort (relative to the effort to maintain and extend the current application) DEI itself might make the application maintenance and extend its functionality in the future.

Keywords

“Communication” “Development agile” “Framework” “PHP” “School” “Software”
“Timetable” “University” “Web application”

Resumo

Este projecto está inserido no âmbito da disciplina de Dissertação/Estágio do ano académico 2008/09 pertencente ao Mestrado em Engenharia Informática (MEI) do Departamento de Engenharia Informática (DEI) da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC).

Guardadas algumas particularidades, a tarefa de elaborar horários académicos dos cursos e depois comunicar a toda a comunidade de pessoas do DEI não tem sido diferente do que ocorre em outros departamentos universitários ao redor do mundo. Visto que a organização manual de aulas, disciplinas, cursos, professores e salas, numa grelha semanal de horários, é algo que consome uma quantidade apreciável de tempo, as vantagens de um sistema informático adequado são evidentes.

Assim, o principal objectivo deste projecto foi o de melhorar o desempenho das tarefas de elaborar e comunicar o horário dos cursos oferecidos pelo Departamento. Com base numa análise do problema e o actual Sistema Informático, foi desenvolvida uma nova aplicação, denominada TTSv2, que está pronta para a elaboração e comunicação dos horários no próximo ano académico.

O autor deste documento tentará convencer o leitor de que a aplicação TTSv2 representa uma evolução relativa ao estado actual, visto incluir melhorias como: **Segurança** – as comunicações entre o servidor Web e os navegadores dos clientes ocorre com os protocolos seguros SSL/TLS. Utilizadores agora têm credenciais para aceder à aplicação e realizar as tarefas; **Menos intervenção manual** – a quantidade de intervenção manual que ocorria anteriormente foi reduzida, tornando o processo mais eficiente; **Disponível para utilização** – a aplicação está disponível online para utilização simultânea por toda a comunidade do DEI, e pode ser acedida de qualquer sítio através do URL <https://timetable.dei.uc.pt/timetable/>; e **Qualidade da manutenção e possibilidade de extensão** – o projecto seguiu o “design patterns” Modelo Vista Controlador (MVC) e recomendações da Framework adoptada. Assim acredita-se que com menos esforço (em relação ao esforço de manter e estender a aplicação actual) o próprio DEI poderá fazer a manutenção da aplicação e estender as suas funcionalidades no futuro.

Palavras-chave

“Aplicação Web” “Comunicar” “Elaborar” “Framework” “Horário Escolar” “PHP” “Software” “Timetable”

Definições e Acrónimos

Aplicação – programa de computador construído em determinada linguagem, que serve ao propósito de realizar tarefas de apoio ao trabalho das pessoas.

CRUD – são as operações para adicionar (Create), exibir (Read), alterar (Update) ou excluir (Delete) dados.

DBMS – Database Management Systems ou sistema gestor de base de dados.

DEI – Departamento de Engenharia Informática da FCTUC.

FCTUC – Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

HC – hard constraints. Restrições que devem ser atendidas, obrigatórias.

HW – Hardware.

MEI – Mestrado em Engenharia Informática da FCTUC.

MVC – Model View Controller

SC – Soft constraints. Restrições que podem ou não ser atendidas, facultativas.

SCRUD – são as operações para fazer pesquisa (Search), adicionar (Create), exibir (Read), alterar (Update) ou excluir (Delete) dados

SI – Sistema Informático. É o conjunto de tarefas, pessoas, aplicações, hardware e comunicações. Termo utilizado tanto no singular como no plural.

SW – Software. Linhas de código arranjadas de forma que compõem uma aplicação.

TTSv2 – TimeTable System version 2. Nome da nova aplicação de gestão de horários do DEI.

UC – Universidade de Coimbra.

WoC – “Web on Campus”. Nome de uma das aplicações utilizadas pelo DEI.

Agradecimentos

Nadja Tamarindo Correia, Pedro Tamarindo Correia, Leone Correia, Ahilson Correia, Raquel Gutierrez Correia, Bruno Correia, Amélia Correia, André Correia, Nina Perez, José Gualberto Almeida, Maria Madalena Pereira, Alcides Tamarindo, Miriam Tamarindo, Lauro Guedes, Sebastião Rildo Diniz, Márcia Kirzner, Fred Kirzner, José Almir Cirilo, Catarina Santiago, Eugénio Santiago, José Alfeu Marques, Filipe Araújo, ao pessoal do CEFET Petrolina, Lilian Amazonas, Marcos Amazonas, Irmãos da Igreja Baptista de Coimbra, aos colegas Estudantes, Professores e Funcionários do DEI / FCTUC, à Comunidade de Países de Língua Portuguesa (CPLP).

Capítulo 1. Introdução

1.1 Âmbito

Este projecto está inserido no âmbito da disciplina de Dissertação/Estágio do ano académico 2008/09 pertencente ao Mestrado em Engenharia Informática (MEI) do Departamento de Engenharia Informática (DEI) da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC).

Guardadas algumas particularidades, a tarefa de elaborar horários académicos dos cursos e depois comunicar a toda a comunidade de pessoas do DEI não tem sido diferente do que ocorre em outros departamentos universitários ao redor do mundo. Visto que a organização manual de aulas, disciplinas, cursos, professores e salas, numa grelha semanal de horários, é algo que consome uma quantidade apreciável de tempo, as vantagens de um sistema informático adequado são evidentes.

Este estágio aconteceu no Grupo de Software and Systems Engineering, Centro de Informática e Sistemas da Universidade de Coimbra (CISUC) no Departamento de Engenharia Informática, no qual o Engenheiro Alexandre Correia foi o Orientando, o Professor Doutor Filipe Araújo foi o Orientador, o Professor Doutor Henrique Madeira e o Professor Doutor Pedro Bizarro integraram o Júri.

1.2 Objectivos

Para o DEI, o objectivo este estágio foi de Melhorar o SI utilizado na tarefa de elaborar e comunicar o horário dos cursos oferecidos pelo Departamento.

Para o Orientando, este estágio teve o objectivo de consolidar os conceitos aprendidos ao longo do curso de Mestrado, com destaque para:

- 1 Desenvolvimento de SW utilizando o paradigma do ciclo de vida iterativo e incremental;
- 2 Implementação de aplicação do tipo Web based;
- 3 Modelização de Base de dados e DBMS;
- 4 Design patterns e Frameworks MVC;

1.3 Resultados

Para atender os objectivos mencionados, com base numa análise do problema e do actual SI, foi desenvolvida uma nova aplicação do tipo Web based para a elaboração e comunicação do horário académico dos cursos do DEI.

A nova aplicação (denominada TTSv2) está pronta para a elaboração e comunicação dos horários no próximo ano académico (resta realizar a formação dos

utilizadores) e acredita-se que ela representa uma evolução, visto agregar melhorias como:

- 1 Segurança – as comunicações entre o servidor Web e os navegadores dos clientes ocorre com os protocolos seguros SSL/TLS. Os utilizadores agora têm credenciais para aceder à aplicação (excepto à área pública) e realizar as tarefas;
- 2 Menor nível de intervenção manual – algumas das intervenções manuais que ocorriam anteriormente foram removidas de forma que o processo tornou-se mais eficiente e com menor possibilidade de erros;
- 3 Disponível para utilização – a aplicação está disponível online para utilização simultânea por toda a comunidade do DEI, e pode ser acedida de qualquer sítio através do URL <https://timetable.dei.uc.pt/timetable/>;
- 4 Qualidade da manutenção e possibilidade de extensão – o que foi implementado seguiu o “design patterns” Modelo Vista Controlador (MVC) e recomendações da Framework adoptada. Acredita-se que é possível para próprio DEI fazer a manutenção da aplicação e estender as suas funcionalidades no futuro;
- 5 Maior nível de fiabilidade – para além do que foi dito antes, todos os dados estão aos cuidados de um Sistema Gestor de base de dados (DBMS).

1.4 Organização do documento

Este documento foi estruturado da seguinte forma:

1 Introdução: pretende fazer o leitor perceber o que será discutido ao longo dos demais capítulos; descreve o âmbito, os objectivos e os resultados alcançados neste estágio.

2 Estado da arte: apresenta os fundamentos da elaboração de horários de cursos, as forças e fraquezas de duas aplicações de utilização gratuita de elaboração e comunicação de horários; faz uma avaliação subjectiva destas opções inclusive comparando a opção de desenvolver uma aplicação específica; e finalmente, apresenta a forma actual como o DEI está a elaborar e comunicar os horários.

3 Proposta do projecto: apresenta o conceito da nova aplicação, as funcionalidades ao nível de negócio e também a sua arquitectura funcional.

4 Metodologia de abordagem: apresenta a estratégia utilizada ao nível de Engenharia de SW; o diagrama de Gantt com as tarefas e prazos previstos e realizados e um breve comentário sobre as principais dificuldades encontradas, também lista os artefactos entregues ao cliente; e finalmente, os componentes utilizados para construir a nova aplicação.

5 Trabalho realizado: apresenta um breve comentário sobre os requisitos e questões relacionadas ao nível de design, nomeadamente as classes, actores, casos de usos e base de dados, utilizados para perceber como a aplicação foi implementada.

6 Conclusões e perspectivas futuras: Este capítulo apresenta as principais conclusões sobre o que foi este estágio, alguns dos resultados esperados; e ações que poderão ser realizadas no futuro para estender ou melhorar o que foi realizado até aqui.

7 Referências: Este capítulo apresenta as referências utilizadas para realização deste estágio. Todas as referências foram consultadas no período de 02 de Abril a 10 de Julho de 2009.

8 Anexos: apresenta as especificações dos requisitos do cliente na forma de “Product backlog”; o documento de especificação de testes da aplicação; o diagrama da base de dados; e finalmente, o manual do utilizador.

Capítulo 2. Estado da arte

Este capítulo apresenta os fundamentos da elaboração de horários de cursos, as forças e fraquezas de duas aplicações de utilização gratuita de elaboração e comunicação de horários; faz uma avaliação subjectiva destas opções inclusive comparando a opção de desenvolver uma aplicação específica; e finalmente, apresenta a forma actual como o DEI está a elaborar e comunicar os horários.

2.1 Sistemas de gestão de horários dos cursos

2.1.1 Elaborar o horário

A elaboração de um horário consiste em alocar um conjunto de objectos, que representam determinados recursos, em espaços temporais pré-definidos de forma a satisfazer uma série de restrições e que sejam contemplados o maior número de necessidades 7.

Com base neste conceito, são encontrados diversos tipos de aplicação prática que tratam da elaboração de horários, como por exemplo para: (i) cursos com suas diversas disciplinas, salas de aulas e professores; (ii) utilização de equipamentos públicos a partir das necessidades dos seus utentes; (iii) mão-de-obra e maquinaria na construção civil.

Para o âmbito deste projecto, foi abordado exclusivamente o problema da elaboração de horários dos cursos oferecidos pelo DEI da FCTUC.

2.1.2 Elaborar e comunicar o horário dos cursos

Adequando o conceito apresentado antes (secção 2.1.1) ao contexto deste estágio, temos que: elaborar um horário (timetable) consiste em alocar um conjunto de objectos (períodos, cursos, disciplinas, aulas) que representam determinados recursos (salas, professores e seus assistentes) em espaços temporais pré-definidos (slots) de forma a satisfazer um conjunto de restrições e a atender o maior número de necessidades (pode ser pensado como a configuração que gera o menor grau de insatisfação da comunidade de pessoas submetidas ao horário).

Depois de elaborados, os horários precisam ser comunicados à comunidade de pessoas que necessitam de saber que estão em vigor. A comunicação ocorrerá da forma habitual no departamento, ou seja, será enviado e-mail da equipa de gestão do DEI para a comunidade de pessoas a indicar que os novos horários estão disponíveis para consulta na área pública do sítio Web da aplicação.

Assim, em função do grande número de instâncias dos objectos, recursos, espaços temporais, restrições (secções 3.2.1 e 3.2.2) e necessidades que costumam fazer parte do processo de elaboração e comunicação de horários, a adopção de SI podem ser decisivos como ferramenta de auxílio na realização desta tarefa.

2.2 Apresentação de duas opções existentes

Está disponível na Internet um generoso conjunto de aplicações para elaboração e gestão de horários escolares 7. Elas estão disponíveis sob licenças de utilização gratuitas ou pagas; voltadas para atender necessidades em instituições onde a combinação de instâncias dos objectos, restrições e objectivos chegam ao máximo de poucas centenas 7 ou para casos onde o número de instâncias facilmente ultrapassa cem vezes este valor 7; implementados com as tecnologias mais variadas (como aplicação Web, para redes locais ou “stand alone”. Em Java, C++, PHP, .NET, etc.).

Assim, parece ser oportuno antes de qualquer decisão, investigar sobre algumas destas existentes aplicações e tentar perceber se é possível aproveitá-las para atender os requisitos (secção 5.1) do DEI.

Embora o número opções seja grande, por um lado, a maior parte delas parece inadequado por fugir ao propósito do DEI, seja porque: (i) é necessário comprar uma licença para utilização como acontece com as aplicações “aSc Timetables” 7 e “Scientia” 7; (ii) está a se dispersar quando aborda mais outras questões em detrimento da elaboração da grelha de horários como acontece com a aplicação “School Tool” 7; e (iii) é mantido por uma única pessoa como acontece com a aplicação “Open Admin for Schools” 7.

Por outro lado, as opções restantes guardam um conjunto de características semelhantes de forma que foi possível classificá-las em efectivamente dois grupos: (i) grupo da aplicação “UniTime” e semelhantes como por exemplo “TimeFinder” 7; e (ii) grupo da aplicação “Fet” e semelhantes como por exemplo “intime”. 7.

Mas adiante será apresentada uma opção de cada grupo, ou seja, as aplicações UniTime (secção 2.2.2) e Fet (secção 2.3.2).

2.2.1 Parâmetros para avaliar as opções

A partir dos requisitos e as informações fornecidas pelo “Product Owner”, foram estabelecidos como referencial de avaliação os seguintes parâmetros:

- 1 Tipo de licença – deve ser gratuito e se assentar em componentes (servidor http, base de dados, Framework, etc.) gratuitos;
- 2 Sistema operativo – deve estar disponível (a funcionar tanto no servidor como nos clientes) para os actuais e mais populares sistemas operativos (Windows, Linux e Mac);
- 3 Instalação e configuração – seja o mais rápido e simples possível, pois para algumas aplicações a instalação e configuração para adequar às necessidades pode representar um esforço maior do que fazer o horário de outra forma, como por exemplo utilizando grelhas Excel;
- 4 Idade e popularidade da aplicação – porque devido à natureza operacional deste assunto, quão mais velha a aplicação e maior a quantidade de utilizadores, melhor as possibilidades de encontrar documentação e fontes de casos de uso;

- 5 Forma de elaboração do horário – deve permitir primordialmente a elaboração do horário através de procedimento manual, porque a configuração para utilizar o procedimento automático pode dificultar a elaboração, haja vista a dinâmica de mudanças de pessoal do DEI;
- 6 Integração com demais aplicações – porque é indispensável que seja capaz de se comunicar (data interchange) com as demais aplicações, por exemplo o “Web on Campus” (WoC) 7, utilizadas pelo DEI.
- 7 Formas de suporte técnico – porque para além da documentação, pode haver a necessidade de esclarecer dúvidas ou até formação para os utilizadores.
- 8 Estar disponível como aplicação Web – porque deve permitir que a equipa de gestão do DEI (de forma colaborativa) que elabora e faz a manutenção do horário possa fazer de qualquer sítio. E mais, à medida que as mudanças na elaboração do horário acontecem, deve permitir a comunicação imediata para a comunidade de utilizadores;
- 9 Reutilizar as instâncias dos objectos, recursos, restrições e necessidades através de períodos diferentes – porque pode poupar grande parcela de tempo quando se evita ter que registar novamente tais instâncias se elas permanecem as mesmas de um período para o outro;
- 10 Flexível ao negócio – parece óbvio, mas não custa lembrar que deve ser flexível ao ponto de o próprio DEI esteja à vontade para adequá-la quando ocorrerem mudanças nas regras do negócio.

Para cada parâmetro de avaliação foi atribuído um peso (de um a três) conforme a importância do parâmetro no projecto. O peso será multiplicado por um valor numa faixa que varia de “um” (quando a opção da aplicação atende plenamente) a “zero” (quando não atende). Ao final somam-se as notas de cada opção e aquela que obtiver o maior resultado será, supostamente, a mais adequada.

Denotar que os parâmetros terão em sua maioria uma atribuição (dos pesos e notas) subjectiva, pois mesmo nos casos onde se consegue quantificar (por exemplo parâmetro 4) a indicação do maior número de utilizadores, não necessariamente conclui que uma opção é mais popular que outra, pois seria necessário medir qual a representação de cada população. O número de utilizadores (expressos por downloads por tempo que a aplicação existe) poderá informar se a ferramenta é popular (quando este número for grande) ou não (quando for pequeno, digamos cem vezes menor que o número anterior).

Outro aspecto que poderia conferir maior qualidade da avaliação seria o de utilizar as ferramentas por um período de tempo, o que não foi realizado. Em outras palavras, a pontuação atribuída foi com base na consulta de suas respectivas documentações.

2.2.2 UniTime

É uma aplicação para elaboração de horários de forma que é capaz de modelar e resolver uma ampla escala de problemas de optimização de horários. É capaz de se adequar desde um pequeno departamento até uma extensa rede distribuída de campus universitários 7.

Para além de tratar do problema de horário dos cursos (course timetable) também trata da solução de gestão de outros aspectos como: horários dos exames; horários de eventos; distribuição dos alunos em turmas conforme as suas preferências, capacidades e recursos de salas de aulas.

Esta aplicação foi concebida a partir das necessidades de gestão dos horários da Purdue University, Indiana USA. Tem a arquitectura implementada em JAVA (Java SE ou Java EE, RMI) 7 e 7:

- 1 É “Open Source”, sob licença GNU GPL 7.
- 2 Está disponível para Windows, Linux e Mac.
- 3 O procedimento de instalação se resume em: (i) instalar o kit de desenvolvimento JAVA SE; (ii) instalar o servidor de container Apache Tomcat; (iii) instalar o servidor de base de dados MySQL; (iv) descompactar os ficheiros da aplicação; e (v) ajustar alguns parâmetros para que esteja pronto a funcionar 7. Em relação a configuração, há um manual online 7 que está dividido em 7 secções (1.1 Course Timetabling, 1.2 Student Sectioning, 1.3 Examination Timetabling, 1.4 Event Management, 1.5 User Preferences, 1.6 Administration, 1.7 Help). Para atender aos actuais requisitos do DEI, faz-se necessário tomar conhecimento das secções 1.1, 1.5, 1.6 e 1.7 do manual online.
- 4 Não foi encontrado no sítio Web do projecto o ano de criação da aplicação como também qualquer informação sobre a comunidade de utilizadores. Percebeu-se porém que as primeiras apresentações sobre esta aplicação ocorreram por volta do ano de 2002 7.
- 5 Permite a elaboração do horário completamente automatizada; completamente manual; ou de uma forma intermédia, ou seja, processo automatizado mas que permite intervenção de quem está a elaborar o horário.
- 6 Permite o intercâmbio de dados através de ficheiros XML 7.
- 7 É possível esclarecer dúvidas por e-mail. A equipa que implementou a aplicação vende o serviço de formação para utilizadores.
- 8 É Web based.
- 9 Reutiliza todos os objectos através dos períodos.
- 10 É flexível ao ponto de permitir que o próprio DEI altere os parâmetros quando da mudança das regras do negócio.

2.2.3 FET

É uma aplicação para elaboração de horários, capaz de se adequar a diversos tipos de escolas e universidades. Depois de lançados os dados e configurados os parâmetros, o fabricante promete que é capaz de gerar o horário em até 20 minutos 7.

Esta aplicação foi concebida na Roménia e actualmente conta com uma pequena equipa de colaboradores. Está disponível para pouco mais de dez idiomas, contudo ainda não para o português. Tem a arquitectura implementada em C++, Qt Framework 7.

- 1 É “Open Source”, sob licença GNU GPL 7.

- 2 Está disponível para Windows, Linux e Mac.
- 3 O procedimento de instalação se resume em executar o ficheiro de instalação (no caso para Windows) e seguir o procedimento. Tem disponível um guia do utilizador online 7.
- 4 Não foi localizado no sítio Web do projecto o ano de criação da aplicação, como também qualquer informação sobre a quantidade de utilizadores. Na lista de discussão há 145 inscritos e a lista foi criada em Maio de 2005.
- 5 Permite a elaboração do horário completamente automatizada; completamente manual; ou de uma forma intermédia, ou seja, processo automatizado mas que permite intervenção de quem está a elaborar o horário.
- 6 Permite o intercâmbio de dados através de ficheiros XML 7.
- 7 É possível esclarecer dúvidas através da colaboração dos membros da lista de discussão 7.
- 8 Não é Web based, porém é possível exportar os relatórios gerados para ficheiros HTML e em seguida publicá-los na Web.
- 9 Reutiliza todos os objectos através dos períodos.
- 10 É flexível ao ponto de permitir que o próprio DEI altere os parâmetros quando da mudança das regras do negócio.

2.3 Fraquezas das opções apresentadas

Após a apresentação das principais características das opções das aplicações, é o momento de comentar as respectivas fraquezas, inclusive é oportuno indicar aqui as fraquezas da opção de implementar uma aplicação específica para o próprio DEI, no qual será denominada **self**.

2.3.1 UniTime

A principal fraqueza encontrada se refere à elevada quantidade de parâmetros que a aplicação apresenta justamente para poder atender um amplo conjunto de necessidades diferenciadas. Assim será necessário maior esforço para perceber quais os parâmetros que devem ser ajustados para atender as necessidades do DEI.

Um exemplo pode ilustrar melhor o que foi dito antes. Os requisitos do DEI para utilização das salas são que estejam disponíveis os campos nome e comentário (por exemplo Nome: E4.7 Comentário: Sala Master Soft Eng.) e que uma sala (Room) não possa ser ocupada por mais de uma aula (Currentclass) durante um mesmo período de tempo (Slot).

O manual do UniTime revela que a quantidade de opções e parâmetros que devem ser observados (Tipos de sala: aulas, laboratórios de computadores, para utilização adicional, para utilização especial, localizadas fora das instalações; capacidade de alunos; a qual grupo de salas ou departamentos pertence; que lista de equipamentos tem; preferência de utilização; etc.) é excessivo e portanto de difícil configuração.

Ilustração 1 – Edição do recurso sala (Room) 7.

Room Detail

EDUC 101 [Edit Room](#) [Edit Room Availability](#) [Edit Room Preference](#) [Edit Room Gr...](#)

Capacity: 4
 Exam Seating Capacity: 2 (Final & Midterm Examinations)
 Controlling Department: 0100 - Central Office
 Coordinates: 1, 1
 Ignore Room Checks: false
 Type: Classrooms
 Groups: Classroom
 Global Features: Chalkboard >= 20 Ft.
 Computer
 Computer Projection
 Fixed Seating

Room Availability

Workdays × Daytime

From: To:	Mon	Tue	Wed	Thu	Fri
7:30a 8:00a	0100-Central Office	0100-Central Office	0100-Central Office	0100-Central Office	0100-Central Office
8:00a 8:30a	0100-Central Office	0100-Central Office	0100-Central Office	0100-Central Office	0100-Central Office
8:30a 9:00a	0100-Central Office	0100-Central Office	0100-Central Office	0100-Central Office	0100-Central Office

Legend:
 0100-Central Office
 0101-Student Instructional Planning
 Not Available
 Free For All

Para os demais objectos (períodos, cursos, disciplinas, aulas) e recursos (professores e seus assistentes) o manual parece manter o excesso de opções.

Numa palavra, a impressão que se tem é que o UniTime seria uma opção adequada se toda a FCTUC resolvesse adoptar esta opção para a elaboração dos horários, o que pode ser razoável, uma vez que o Departamento de Engenharia Electrotécnica e Computadores (DEEC) comprou uma aplicação de horários, o DEI já anda a investir tempo em busca de uma opção e outros departamentos terão seguramente problemas semelhantes. Neste caso o esforço em preparar uma equipa para investigar com detalhe esta opção poderia ser compensador.

Para o âmbito das necessidades actuais do DEI (e de um futuro não muito distante) parece ser uma escolha com um excesso de opções que acaba por complicar a gestão dos horários.

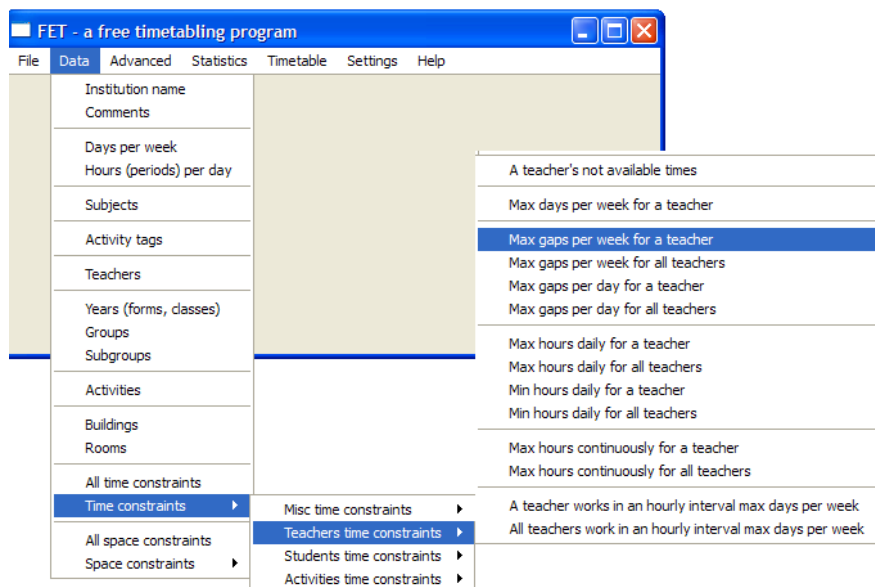
2.3.2 FET

Esta opção suporta múltiplos idiomas e a quantidade de parâmetros de configuração é bem menor, e portanto, pode estar mais próximo da realidade do DEI. As fraquezas são:

- 1 Não ser Web based e portanto não atende ao parâmetro de avaliação 2.2.1.
- 2 É necessário algum esforço (embora pareça menor que para a aplicação anterior) para perceber a estrutura de parâmetros de configuração da aplicação.

Por fim a impressão que se tem é que o FET seria uma opção adequada, contanto que o horário seja elaborado em um computador “stand alone” e depois através de procedimento manual exportar as diversas grelhas de horário (no último semestre foram geradas por volta de 160 diferentes grelhas de horários) para o formato HTML, para finalmente poder publicá-las na Web.

Ilustração 2 – alguns parâmetros do FET timetable software



2.3.3 Aplicação self

Do que foi exposto antes fica evidente que se o DEI decidir utilizar uma das opções propostas terá que dedicar horas de esforço para perceber a parametrização da aplicação, afim de ajustá-la às suas necessidades. E mais, mesmo depois deste esforço poderão surgir situações que podem acabar por condicionar o DEI a se adequar à aplicação (e não o contrário) uma vez que os limites de flexibilidade de aplicações generalistas costumam ser menores que os limites de flexibilidade de uma aplicação específica para o próprio DEI.

É bem verdade que uma aplicação desenvolvida para o próprio DEI (self) tem a possibilidade de atender o maior conjunto de necessidades, porém é importante atentar para as seguintes potenciais fraquezas:

- 1 O facto de ao modelar a aplicação conhecendo os requisitos voltados exclusivamente para o DEI, faz com que a equipa de desenvolvimento acabe por embutir todo o conjunto de regras. Isto numa primeira impressão é muito confortável para a comunidade de utilizadores, pois “não precisam parametrizar praticamente nada”, todas as regras do negócio são “built-in”, pelo que só é preciso lançar os dados e já está a gerar os resultados esperados. Mas e se as regras de negócios mudarem no futuro? No mundo real elas costumam mudar! E agora o que fazer se a equipa que desenvolveu a aplicação foi desfeita ao término do projecto? Mesmo que se reúna uma nova equipa, tentar perceber o código costuma ser uma missão mais dispendiosa do que perceber o manual de configuração de alguma aplicação.
- 2 Enquanto que nas opções anteriores o guia do utilizador e qualquer outra fonte de documentação (por exemplo fórum, comunidade de utilizadores) já estão estabelecidos e disponíveis, no caso de uma solução self isto ainda será implementado e a comunidade de utilizadores só estará numa fase proveitosa depois de alguns ciclos de trabalho em torno da aplicação (no caso, alguns semestres).

2.4 Resultados da avaliação

A Ilustração 3 resume os parâmetros de referência com o resultado da avaliação das opções.

Ilustração 3 – Resultado da avaliação

Parâmetro avaliação	Peso	UniTime	Nota	FET	Nota	Self	Nota
Licença gratuita	3,0	GNU GPL	1,0	GNU GPL	1,0	(*1)	1,0
Sistema operativo	3,0	Win Lin Mac	1,0	Win Lin Mac	1,0	Win Lin Mac	1,0
Dificuldade instalação e configuração	3,0	Maior	0,0	Intermédia	0,5	Mínima	1,0
Idade/ popularidade/ guia do utilizador	2,0	2002/ -/ sim	0,9	2005/ 145 inscritos no fórum/ sim	1,0	2009/ -/ não	0,0
Horário (automático; manual; interactivo)	1,0	Sim/ sim/ sim	1,0	Sim/ sim/ sim	1,0	Não/ sim/ não	0,5
Data interchange	1,0	XML, PDF	1,0	XML, HTML	1,0	TXT, PDF (*2)	1,0
Suporte técnico pelo fabricante	2,0	Vende formação. E-mail gratuito.	1,0	Colaboração do fórum, gratuito.	0,5	(*3)	0,8
Web based	3,0	Sim	1,0	Não	0,0	Sim	1,0
Reutilização objectos	1,0	Sim	1,0	Sim	1,0	Sim	1,0
Flexível ao negócio	1,0	Parâmetros	0,9	Parâmetros	0,9	Código	1,0
Total	20,0		16,7		14,4		17,1

(*1) A aplicação é do DEI, portanto ele decidirá o que será feito. (*2) É possível utilizar o navegador para exportar os dados. Para importação poderá ser implementado se requisitado. (*3) Cabe ao DEI decidir como será o suporte.

A análise da Ilustração 3 revela uma subtil vantagem em favor do desenvolvimento de uma aplicação (self) específica para o DEI.

No entanto, há dois aspectos cruciais que pesam contra qualquer uma das aplicações anteriormente apresentadas: a UniTime é excessivamente complexa; enquanto que a Fet não é Web based. Só por si estes factos são suficientemente motivantes para iniciar a tarefa de criar uma nova aplicação.

2.5 Sistema actual do DEI

2.5.1 Caracterização

A actual aplicação se propõe a permitir que seja elaborada a grelha de horários do DEI a partir das demandas de cursos, disciplinas, tipos de aulas, preferências de horários dos professores, salas de aula, – entre outros.

Assim, para cada novo semestre a partir das disciplinas que serão ofertadas em cada curso, deverão ser escolhidos as salas de aula e os professores de forma a satisfazer um conjunto de regras.

Actualmente a aplicação é acedida por um único utilizador, que é a pessoa que elabora a grelha de horário.

2.5.2 Arquitectura

A elaboração do horário é realizada por uma única pessoa através de um processo de quatro etapas:

1. A pessoa que elabora o horário inicia a aplicação escrita em JSF (o acesso é feito através do URL <http://filipius.dei.uc.pt:8080/Timetables/faces/index.jsp>) e lança os principais objectos e recursos (que equivalem à secção 3.2.1) e gera um conjunto de ficheiros texto plano com o conteúdo das grelhas elaboradas e demais dados;
2. Depois ela realiza uma série de operações manuais de rename, copy & paste dos ficheiros gerados, para que se adequem a um standard para utilização na próxima etapa;
3. Por fim ela, inicia Microsoft Excel e, para os ficheiros arranjados na etapa anterior, corre um conjunto de macros para ter como resposta as grelhas Excel prontas para impressão.
4. Finalmente, o procedimento de lançar as grelhas de horário no WoC é realizado de forma manual, ou seja, um funcionário do DEI tem à mão os ficheiros texto plano e então vai lançando no WoC os horários grelha a grelha.

2.5.3 Fraquezas

As principais fraquezas encontradas na arquitectura da aplicação actual foram:

1. Significativa intervenção manual – entre cada uma das etapas há grande intervenção manual. A nova aplicação se propõe a transformar as quatro etapas em apenas duas. Pois pretende eliminar as etapas 2 e 3 e para a etapa 4 (secção 2.5.2) pretende que a troca de dados aconteça de forma a sofrer a menor intervenção manual desejável.
2. Ausência de Sistema Gestor de Base de dados (DBMS) – tem a sua base de dados assentada em ficheiros texto no qual a aplicação em JSF controla as regras de definição e manipulação dos dados. Este cenário deixa a aplicação JSF mais complexa (pois precisa implementar as regras que tratem os dados) e perde a oportunidade de utilizar alguns serviços que um DBMS poderia proporcionar, como por exemplo: integridade referencial dos dados das tabelas; indexação; campos de incremento automático; gatilhos (triggers); etc.
3. Autenticação e autorização – não existem.
4. Alteração on the fly – depois de pronta a grelha de horários, qualquer mudança desejada, vai provocar a repetição de todo o processo.
5. Disponibilidade – não está disponível à comunidade de utilizadores para consulta imediata.
6. Utilização simultânea – não é possível para a comunidade de utilizadores (seja qual for o papel) utilizar a aplicação em simultâneo, pois visto não haver políticas de autorização, no momento em que um funcionário do DEI está a elaborar o horário, é possível para o aluno que aceder a aplicação, ter o poder de desfazer a acção do funcionário.
7. Localizar dados – há dificuldade em fazer pesquisa (search) nos dados. Não há uma forma, durante uma sessão de trabalho, de isolar o conjunto dos dados que o utilizador está interessado em visualizar. Por exemplo em quais períodos aparecem uma determinada disciplina?

8. Pouco User friendly – as interfaces com o utilizador poderiam ser melhoradas, visto por exemplo ser uma tarefa não muito trivial ao utilizador com conhecimentos básicos de informática (pessoa da equipa de gestão do DEI) editar ficheiros em texto plano para ajustar algum detalhe quando ocorrerem erros nas macros. Outro exemplo, não é trivial para este mesmo utilizador fazer um ajuste, como por exemplo, incluir uma aula para o horário entre 13 e 14 horas.

2.5.4 Módulos

As funcionalidades da aplicação são definidas a partir de um conjunto de módulos, estes são invocados a partir de chamadas na interface do utilizador que é orientada por um menu principal. A seguir os principais módulos da aplicação.

2.5.4.1 Add subject

Permite visualizar as existentes e adicionar uma nova disciplina (Subject). Por exemplo: SUB1TESTE.

2.5.4.2 Delete subject

Permite remover uma disciplina (Subject). Não é possível remover a disciplina se ela estiver relacionada com algum Curso-ano.

2.5.4.3 Add new name to subject

Permite visualizar os existentes e adicionar um nome alternativo para uma disciplina (Subject). Por exemplo: para a disciplina CAD podemos incluir HPC, ou Computação de Alto Desempenho ou ainda High Performance Computing.

2.5.4.4 Delete name to subject

Permite remover um nome alternativo de uma disciplina (Subject).

2.5.4.5 Add Course-Year

Permite visualizar os existentes e adicionar um nome de curso-ano. Por exemplo: MEI2009TESTE;

2.5.4.6 Delete Course-Year

Permite remover um nome de curso-ano.

2.5.4.7 Add & Delete Subject to Course-Year

Ilustração 4 – Curso ano

Course-Year

The screenshot shows a web interface for managing subjects for a specific course-year. At the top, a dropdown menu is set to 'MEI2009TESTE'. Below this, there are two columns of subject lists. The left column, titled 'Available Subjects', contains a list of subject codes: AED, ATD, MIEBM-ADAR, API, C, CG, CE, CAD, DC, and --. The right column, titled 'Current Subjects', contains a list of subject codes: SUB1TESTE, SUB2TESTE, and SUB3TESTE. Between these two columns are two buttons: 'Add' and 'Delete'. At the bottom of the interface is a 'Done' button.

Permite adicionar ou remover disciplinas para um Curso-ano. Por exemplo: para o curso-ano MEI2009TESTE é possível adicionar as novas disciplinas a partir de um conjunto disponível (na lista Available Subjects) ou remover as disciplinas que constam disponíveis (Current Subjects) para o respectivo Curso-ano.

2.5.4.8 Add Professors

Permite visualizar os existentes e adicionar um professor (login e nome). Por exemplo: Para o professor Joaquim Silva pode ser informado login e nome como JS e Joaquim Silva.

2.5.4.9 Delete Professors

Permite remover um professor.

2.5.4.10 Add Professors Preferences

Permite visualizar as existentes e adicionar novas preferências de horários (dia da semana, Hora inicial e duração) e o estado de disponibilidade de cada professor. **Este módulo não está a funcionar!**

Ilustração 5 – Preferências do professor**Professor Preferences**BR **Current list of preferences****Add a new preference**

Day (1 - Sun; 7 - Sat)	<input type="text"/>	Hour
<input type="text"/>	Duration	<input type="text"/>
Green <input type="button" value="v"/>		
<input type="button" value="Add"/>	<input type="button" value="Done"/>	

2.5.4.11 Delete Professors Preferences

Permite remover as preferências de horários de cada professor. **Esta funcionalidade não está a funcionar!**

2.5.4.12 Add Type of Class

Permite visualizar os existentes e adicionar novos tipos de aulas (o identificador do tipo; o número de vezes que ocorre por semana; e a duração em minutos). Por exemplo, para um tipo de aula teórica de 1 hora que ocorre 2 vezes por semana pode ser adicionado da seguinte forma: T1H2V, 2, 60.

2.5.4.13 Delete Type of Class

Permite remover um tipo de aula. Não é possível remover o tipo de aula se este estiver relacionado à uma disciplina.

2.5.4.14 Add Class of Subject

Permite visualizar os existentes e adicionar novos tipos de aulas para cada disciplina. Por exemplo, se para a disciplina SUB1TESTE ocorrem os seguintes tipos de aula (tipo Teórica com duração de 2 horas 1 vez por semana; e prática com duração de 2 horas, 1 vez por semana) estes tipos (e outros) podem ser adicionados para a disciplina em questão.

2.5.4.15 Delete Class of Subject

Permite remover os tipos de aula relacionados à disciplina. Não é possível remover o tipo de aula da disciplina.

2.5.4.16 Assign professors

Permite visualizar as existentes e adicionar novas associações de aulas com professores para cada disciplina. Aqui é definido que professor ministra o tipo de aula dentro de uma disciplina. Por exemplo, se a disciplina SUB1TESTE existem aulas do tipo teórica de 2 horas 1 vez por semana (T1) e Prática de horas 1 vez por semana (PL1). Então é possível associar a cada tipo de aula um ou mais professores.

Ilustração 6 – Professores das disciplinas

Teachers of Subjects

IN

Code	Description	All Profs.	Add	Del	Assigned Profs.
<input type="text" value="2576"/>	T1	ADC AJM ALC AM BR CLB EFC	>	<	PG HCM
<input type="text" value="2577"/>	PL1	ADC AJM ALC AM BR CLB EFC	>	<	PG HCM
<input type="text" value="2578"/>	PL2	ADC AJM ALC AM BR CLB EFC	>	<	PG HCM

2.5.4.17 Add Room

Permite visualizar as existentes e adicionar novas salas de aulas (nome; e observação). Por exemplo, para o Anfiteatro A da FCTUC pode ser adicionado da seguinte forma (Anf-A; Anfiteatro com 100 assentos na FCTUC).

2.5.4.18 Delete Room

Permite remover a sala de aula. Gerou uma exceção na tentativa de remover uma sala:

```
javax.servlet.ServletException:
  #{rooms.delData}:
  java.lang.NullPointerException
  javax.faces.webapp.FacesServlet.service(FacesServlet.java:2
  77).
```

2.5.4.19 Add Schedule to Time Table

Ilustração 7 – Adicionar horário[Go To Delete TT](#)

MEI2009TESTE ▾
 SUB1TESTE - T1 PROFTESTE/ADC ▾

Done

[PROFTESTE](#)[ADC](#)**Add To Time Table**

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00		here	here	here	
8:30		here	here	here	
9:00	SUB1TESTE (PL1) RoomTeste PROFTESTE	here	here	here	
9:30	SUB1TESTE (PL1) RoomTeste PROFTESTE	here	here		
10:00	SUB1TESTE (PL1) RoomTeste PROFTESTE	here	here		
10:30	SUB1TESTE (PL1) RoomTeste PROFTESTE	here	here		
11:00	here	here	here		
11:30					
12:00					
12:30					

Permite visualizar as existentes e adicionar novas marcações na grelha de horários, conforme o curso-ano, e o tipo de aula e professor relacionados à disciplina. Por exemplo, para o curso-ano MEI2009TESTE, a disciplina SUB1TESTE que tem aulas do tipo T1 e estão associados os professores PROFTESTE e ADC poderá escolher qualquer das células da grela que contém um link com a palavra here para escolher a sala de aula, no exemplo ROOMTESTE disponível naquele horário.

Ilustração 8 – Adicionar sala no horário

RoomTeste ▾
 Add Cancel

Observations: Sala de testes

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00					
8:30					
9:00	SUB1TESTE (PL1) RoomTeste PROFTESTE				
9:30	SUB1TESTE (PL1) RoomTeste PROFTESTE				
10:00	SUB1TESTE (PL1) RoomTeste PROFTESTE				
10:30	SUB1TESTE (PL1) RoomTeste PROFTESTE				
11:00					
11:30					
12:00					

2.5.4.20 Delete Schedule to Time Table

Permite remover as marcações da grelha de horários, conforme o curso-ano, e o tipo de aula e professor relacionados à disciplina.

2.5.4.21 See Professors Time Table

Permite visualizar a grelha de horários, através da escolha de um professor.

2.5.4.22 See Roons Time Table

Permite visualizar a grelha de horários, através da escolha de uma sala de aula.

Capítulo 3. Proposta do projecto

Este capítulo apresenta o conceito da nova aplicação, as funcionalidades ao nível de negócio e também a sua arquitectura funcional. A nova aplicação será denominada TTSv2, que é o acrónimo de TimeTable System em sua segunda versão, a primeira seria aplicação actual.

3.1 Conceito

Pretende-se desenvolver uma aplicação para elaboração e comunicação de horários (secção 2.1.1) do DEI que permitirá à equipa de gestão, sempre que desejar, elaborar a grelha de horários das disciplinas dos cursos oferecidos pelo departamento. Tão logo o horário esteja pronto a comunicação ocorre de imediato, bastando à comunidade do DEI aceder ao sítio Web e consultar as grelhas sob a forma de relatórios.

Os utilizadores são classificados da seguinte forma:

- 1 Membros da equipa de gestão do DEI – com permissões totais ou parciais sobre as funcionalidades aplicação. Estas permissões irão variar de acordo com o papel que terão;
- 2 Professores e assistentes – com permissões parciais sobre a aplicação; e
- 3 Alunos e demais interessados (Público) – com permissão para aceder a área pública para consulta dos relatórios das grelhas de horários.

A aplicação deve permitir que o ciclo de elaboração e comunicação do horário (ou seja: o registo dos objectos e recursos; a elaboração do horário; e os resultados na forma de grelhas) seja mais eficiente que à forma realizada pela aplicação actual. E mais, as grelhas elaboradas devem estar imediatamente disponíveis (comunicação):

- 1 Na Web para consulta pela comunidade de pessoas interessadas;
- 2 Para impressão dos banners de papel tamanho A4, que são afixados nas portas das salas de aulas ou quadros de avisos; e
- 3 Sob a forma de ficheiros em texto plano ou PDF.

A aplicação deve estar pronta a funcionar em no máximo 90 (noventa) dias, dos quais deve estar pronta para realização dos testes finais de aceitação com os utilizadores em no máximo 75 (setenta e cinco) dias.

3.2 Termos

Como foi dito na secção 2.1.1, elaborar um horário consiste em alocar um conjunto de objectos que representam determinados recursos em espaços temporais pré-definidos (slots) de forma a satisfazer um conjunto de restrições e a atender o maior número de necessidades.

A seguir procurar-se-á esclarecer tais termos, visto que elas representam os fundamentos das funcionalidades que a aplicação deverá ter.

Ao nível de negócio as entidades apresentadas adiante são objectos, em excepção às restrições e aos recursos de salas de aulas, professores e seus assistentes.

Estas entidades ao nível do **negócio** são os fundamentos para os módulos (secção 3.3) ao nível de **arquitectura** (secção 3.4) e para as **classes** (secção 5.2.1) ao nível de **design** (secção 5.2) da nova aplicação proposta para o DEI.

3.2.1 Objectos e recursos

3.2.1.1 Período (Period)

Corresponde a um evento de tempo que ocorre no início de cada ciclo de elaboração do horário do DEI. Antes do início de cada semestre escolar é necessário criar um novo período, como por exemplo 2009-1 que corresponde ao 1 semestre escolar no ano de 2009.

3.2.1.2 Slot

É um conjunto de espaços temporais que compõem a grelha de horários. Em semelhança a uma grelha Excel, cada Slot corresponde a uma célula com endereço de linha correspondente ao número de ordem do slot (onde se exhibe o horário) e endereço de coluna correspondente ao dia da semana. Por exemplo, na ilustração à seguir os slots das 9h as 11 estão ocupados na terças e quartas-feiras todas as semanas no segundo semestre escolar do ano de 2008 (período 2008-2).

Ilustração 9 – Exemplo de grelha com slots

Period: 2008-2 Course: MEI

	MON	TUE	WED
08:30-09:00			
09:00-10:00		EDJ T1 (B1) L Roque	EDJ PL1 (C5.1) L Roque
10:00-11:00		EDJ T1 (B1) L Roque	EDJ PL1 (C5.1) L Roque
11:00-12:00			
12:00-13:00			

3.2.1.3 Curso (Course)

É um conjunto de disciplinas que são apresentadas progressivamente ao longo de alguns períodos para o grupo de alunos vinculados. Por exemplo MEI corresponde ao curso do Mestrado em engenharia Informática.

3.2.1.4 Disciplina (Subject)

É um conjunto de aulas (secção 3.2.1.9) que são apresentadas progressivamente ao longo do período para o grupo de alunos. Por exemplo CAD corresponde à disciplina de “Computação de Auto Desempenho” presente nos cursos de Licenciatura e Mestrado em Engenharia Informática.

3.2.1.5 Tipo de aula (Typeclass)

É a classificação da aula (secção 3.2.1.9) conforme sua natureza, por exemplo T para aulas do tipo teórica, PL para Práticas de Laboratório, TP para Teórico-Práticas, etc.

3.2.1.6 Grupo de utilizador (GroupUser)

É a classificação dos utilizadores pelo tipo de papel que irão desempenhar na aplicação. Por exemplo `administrators` terão permissões totais sobre as funcionalidades da aplicação.

3.2.1.7 Utilizador (User)

É a identificação do utilizador na aplicação, por exemplo um dos funcionários do DEI, com o nome de João da Silva, que terá o papel de administrador (e portanto estará vinculado ao grupo de utilizadores `administrators`) na aplicação, poderá aceder como o nome de utilizador `João Silva` e uma palavra passe criada para esta finalidade.

Utilizadores do tipo `professors` poderão ser vinculados às aulas e informar suas preferências de horários para ministrar as aulas (secção 6.4).

3.2.1.8 Sala (Room)

É a identificação da sala de aula, por exemplo `E4.7` para a sala E4.7 localizada na torre E do prédio do DEI ou `BP-Anf.A` para o anfiteatro A do edifício principal da FCTUC.

3.2.1.9 Aula (Currentclass)

Consiste em transmitir um determinado conteúdo (pertencente a uma determinada disciplina) por isto precisa ocupar slots de tempo durante as semanas do período escolar. Esta actividade é realizada por tipo específico de utilizador (apenas professores ou auxiliares).

É formada pela união de um período, curso, disciplina e tipo de aula. Por exemplo `2009-1 MEI CAD PL1` corresponde à aula Prática de Laboratório 1, da disciplina Computação de Auto Desempenho do curso de Mestrado em Engenharia Informática do 1º semestre do ano de 2009. No exemplo anterior, poderíamos ter uma segunda aula de prática de laboratório (`PL2`) que seria nomeada `2009-1 MEI CAD PL2`.

3.2.1.10 Aula Utilizador (Currentclass_User)

Uma disciplina pode ter um ou mais utilizadores (Professores) que ministram de forma partilhada determinadas aulas, assim é possível vincular a uma aula qualquer número de professores (inclusive nenhum). Por exemplo, poderíamos ter a aula teórica

2009-1 MEI SGD T1. ministrada pelos Professores H Madeira, P Bizarro ou M Vieira.

3.2.1.11 Slot Utilizador (Slot_User)

Cada utilizador pode ter preferências de horários para ministrar as aulas, assim é possível vincular a cada utilizador quais slots prefere ministrar as aulas (secção 6.4).

3.2.1.12 Horário (Timetable)

Quando uma aula é alocada num slot de tempo e utiliza uma sala, se diz que esta aula tem um horário. Assim quando várias aulas passam a ter seus horários, se diz que está elaborada uma grelha de horários. Por exemplo, poderíamos ter a aula teórica 2009-1 MEI SGD T1 ministrada pelos Professores H Madeira, P Bizarro ou M Vieira que na terça-feira ocupa 2 slots (das 11:00 as 12:00 e das 12:00 as 13:00 horas) na sala E4.7. Outra aula 2009-1 MEI CAD T1 ministrada pelo Professor F Araújo que também na terça-feira ocupa 2 slots (das 9:00 as 10:00 e das 10:00 as 11:00 horas) na sala E4.5. Para o preenchimento dos demais horários, segue o mesmo raciocínio.

3.2.2 Conjunto de restrições

As restrições impostas pela elaboração do horário podem ser divididas em dois grupos: hard constraints (HC) ou restrições de natureza obrigatória – são aquelas que o horário só poderá ser elaborado depois que atender a restrição; e soft constraints (SC) ou restrições de natureza facultativa – são aquelas que o horário pode ser elaborado mesmo que não atenda a restrição (muito embora deve ocorrer a notificação de não atendimento). À Seguir as restrições hard:

1. HC1 – uma mesma disciplina quando pertence a mais de um curso terá seu horário comum a todos, e somente todos, os cursos num mesmo período.
2. HC2 – as aulas (seja qual for o tipo) de uma mesma disciplina não podem acontecer num mesmo slot de horário num mesmo período.
3. HC3 – duas aulas diferentes não podem ocupar a mesma sala de aula num mesmo slot de horário num mesmo período.
4. Outras HC – existem outras restrições que parecem ser mais explícitas como: (i) o nome de cada objecto ou recurso deve ser único; (ii) deve existir integridade referencial entre os objectos, por exemplo não ser possível excluir um utilizador se ele está associado a alguma aula; (iii) para qualquer disciplina não pode haver mais slots na grelha do que é suposto o disponível, por exemplo uma disciplina tem disponível dois slots para utilização e seja possível meter na grelha de horários 5 slots.

Finalmente, as restrições soft:

1. SC1 – um utilizador não pode estar em duas aulas diferentes num mesmo slot de horário num mesmo período.
2. SC2 – cada aula deve ter um utilizador associado.

3. SC3 – utilizadores só poderão ministrar aulas no seu respectivo horário preferencial (secção 6.4).

3.3 Funcionalidades (módulos)

Muitas das entidades (apresentadas na secção 3.2.1) ao nível do negócio, representam os módulos ao nível de arquitectura. E mais, muitos destes módulos tem o comportamento semelhante entre si, em relação a forma como manipulam os dados, daí serem denominados aqui como módulos tipo.

3.3.1 Gestão de módulos tipo

Deve possibilitar as operações (de pesquisar “Search”, adicionar “Create”, exibir “Read”, alterar “Update” ou excluir “Delete”) sobre os dados de Períodos, Cursos, Disciplinas, Aulas e os seus tipos, Utilizadores e grupos, Utilizadores para as aulas, Salas, Preferências de utilizadores para slots (secção 5.2.3.2).

3.3.2 Gestão de Slots

Deve possibilitar as mesmas operações apresentadas na secção 3.3.1, e mais, em copiar o conjunto de slots de um período para outro com o menor esforço de acção do utilizador (secção 5.2.3.3).

3.3.3 Gestão de Horários

Deve possibilitar operações semelhantes as apresentadas em 3.3.1, e mais, em disponibilizar as grelhas de horário, sob a forma de relatórios que devem estar prontos para consulta, imediatamente após a elaboração do horário (secção 5.2.4).

3.3.4 Geração de ficheiros para exportação

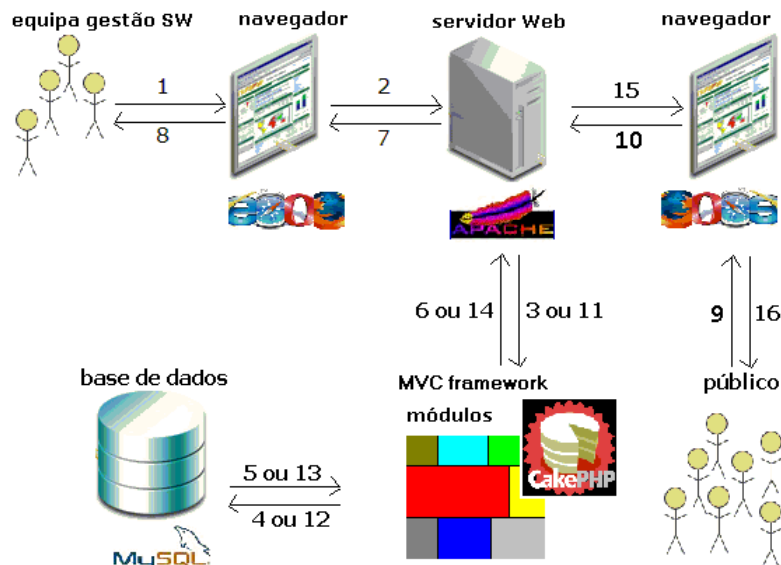
Deve possibilitar a exportação de dados para outras aplicações do DEI (por exemplo o WoC). Isto pode ser feito através do navegador de qualquer pessoa da equipa de gestão, para o formato de texto plano 7 ou PDF 7.

3.4 Arquitectura

A arquitectura da aplicação procura atender aos requisitos ao nível de negócio de forma a ajudar perceber o fluxo da informação (onde começa, como transita e onde termina).

A figura a seguir ilustra o fluxo das mensagens em dois dos principais casos de uso: (i) equipa gestão realiza a elaboração do horário (setas de 1 a 8); e (ii) aluno acede área pública para visualizar o relatório da grelha de horários por curso (setas de 9 a 16).

Ilustração 10 – Arquitectura da aplicação



1. O actor (qualquer membro da equipa de gestão) acede ao sítio Web a partir do seu navegador preferido, realiza uma acção através de uma das Vistas, por exemplo: adicionar uma aula na grelha de horários.
2. O navegador passa a mensagem para o servidor Web.
3. O servidor Web passa a mensagem, para o controlador da Framework MVC, que faz o tratamento lógico da mensagem.
4. Em seguida, o Controlador passa a mensagem para o Modelo que executa uma acção (adicionar a aula) sobre a estrutura de dados na base de dados.
5. Digamos que a acção correu com sucesso, pelo que o Modelo gera a mensagem: aula adicionada na grelha de horários com sucesso. A nova mensagem agora faz o caminho de volta, ou seja, é transmitida do Modelo para o Controlador.
6. A mensagem transita através da Framework MVC até o servidor Web.
7. Do servidor Web até o navegador do actor.
8. O actor ver a nova mensagem: aula adicionada na grelha de horários com sucesso.
9. Os actores do tipo público podem consultar o relatório da grelha do horário e verificar que lá estará a aula adicionada nas etapas anteriores. Para isto acedem ao sítio Web a partir do seu navegador preferido, realizam uma acção através de uma das Vistas, por exemplo: visualizar relatório grelha de horários do curso desejado.
10. Semelhante a 2.
11. Semelhante a 3.
12. Semelhante a 4.
13. Semelhante a 5.
14. Semelhante a 6.
15. Semelhante a 7.
16. Finalmente, o actor ver o resultado que é a grelha de horários do curso desejado.

Capítulo 4. Metodologia de abordagem

Este capítulo apresenta como estratégia utilizada para construção a nova aplicação o paradigma do ciclo de vida Iterativo e Incremental e alguns conceitos oriundos de SCRUM; o diagrama de Gantt com as tarefas e prazos previstos e realizados e um breve comentário sobre as principais dificuldades encontradas, também lista os artefactos entregues ao cliente; e finalmente, os componentes utilizados para construir a nova aplicação.

4.1 Desenvolvimento iterativo e incremental

No contexto deste trabalho, será adoptado o princípio da engenharia de software 7 voltado para o paradigma de desenvolvimento baseado no ciclo de vida iterativo e incremental 7 porque pretende-se que ocorra iterações pontuais com o “Product Owner” e a aplicação seja entregue de forma de evolucionária, ou seja, a cada nova versão sejam agregadas melhorias e novas funcionalidades.

Visto o prazo de entrega (90 dias) serão adoptados alguns conceitos da metodologia de desenvolvimento ágil de SW, denominada SCRUM 7:

- 1 Product Owner – é o dono do produto, ou seja, o cliente, que vai dizer a equipa de desenvolvimento o que ela deve implementar para atender seus requisitos. O DEI (que é o cliente neste projecto) nomeou o Professor Doutor Filipe Araújo como o Product Owner.
- 2 Equipa de desenvolvimento – pessoa (ou grupo) nomeada pelo DEI, para implementar a aplicação. Neste projecto o Engenheiro Alexandre Correia é a equipa de desenvolvimento.
- 3 Product backlog – lista de requisitos que o “product owner” deseja.
- 4 Sprint – o product backlog deve ser dividido em grupos para que a equipa de desenvolvimento possa implementar um grupo em cada ciclo de trabalho. Cada ciclo de trabalho é um Sprint.
- 5 Entrega de Release – ao final de cada Sprint o resultado será uma versão (release) da aplicação, que será apresentada ao “product owner” e tão logo a equipa de desenvolvimento receba o sinal positivo do teste de aceitação, passa a iniciar o próximo Sprint. Para o caso de algo que tenha falhado, o destino será definido em reunião do “product owner” com a equipa de desenvolvimento.
- 6 Reuniões com o “product owner” – no mínimo uma vez por semana para passar o ponto de situação dos trabalhos.

4.2 Plano de trabalho

Aqui, é apresentado o Diagrama de Gantt com o planeamento das tarefas e metas a serem atendidas neste projecto. E mais, um breve comentário das principais dificuldades encontradas daquilo que foi realizado.

Ilustração 11 – Diagrama de Gantt do plano de trabalho

Id	Tarefa	Início	Fim	01/04/09	08/04/09	15/04/09	22/04/09	29/04/09	06/05/09	13/05/09	20/05/09	27/05/09	03/06/09	10/06/09	17/06/09	24/06/09	01/07/09	08/07/09	15/07/09	22/07/09	
1	Regras negócio, perceber e estender sistema actual	01/04/09	30/04/09	o	o	o	o	o													
		01/04/09	07/04/09	X																	
1a	Regras negócio, nova aplicação	08/04/09	30/04/09	o	o	o	o														
		08/04/09	30/04/09	X	X	X	X														
2	Arquitectura, design e BD	01/05/09	15/05/09						o	o											
		01/05/09	27/05/09						X	X	X	X									
3	1º Sprint - módulos das classes tipo	16/05/09	31/05/09								o	o									
		16/05/09	03/06/09								X	X	X								
4	2º Sprint - módulo horário	01/06/09	15/06/09										o	o							
		01/06/09	22/06/09										X	X	X						
5	3º Sprint - módulo slots, Autorização e autenticação	16/06/09	30/06/09												o	o					
		16/06/09	30/06/09												X	X					
6	Escrita relatório	15/06/09	10/07/09												o	o	o	o			
		22/06/09	10/07/09												X	X	X				
7	refinamentos e monitorização 1ª utilização	11/07/09	22/07/09																	o	o

O = planeado, X = realizado

Semana→ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Dias 0 7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112

Tarefa Id 1 – dia 1-30 antes, 1-7 depois: Perceber as regras do negócio, as funcionalidades do SI actual e investigar o código da aplicação para depuração de bugs actualmente existentes e a adição de novas características essenciais.

Foi avaliado o risco de ao tentar continuar a trabalhar com o código da aplicação actual e utilizar um conjunto de componentes que eram conhecidos, mas não “os mais conhecidos” pela equipa de desenvolvimento (por exemplo PostgreSQL 7, Glassfish 7 e JAVA Server Faces 7) o projecto poderia não ficar pronto no tempo disponível. Por isto, resolveu-se encerrar a tarefa Id 1 ao final do sétimo dia e partir para realizar a tarefa Id 1a.

Tarefa Id 1a – dia 8-30: A maior dificuldade aqui foi a ambientação com a Framework MVC, mas superada esta dificuldade, tudo correu dentro do planeado.

Tarefa Id 2 – Dia 31-45: Modelar a base de dados e adequá-la ao standard da Framework, representou um esforço acima do esperado, pelo que foi esta a razão principal das duas semanas de atraso.

Tarefa Id 3 – Dia 46-60: 1º Sprint – módulos das classes tipo. Colocar a funcionar o primeiro módulo da aplicação, foi a maior dificuldade encontrada, vencida esta, o projecto começou a dar sinais de que ficaria pronto no tempo disponível. Denotar que o atraso foi menor que na tarefa anterior, mas ainda assim foi registado atraso de uma semana.

Tarefa Id 4 – Dia 61-75: 2º Sprint – módulo horário. Sem dúvida, foi o módulo mais difícil de implementar, visto que as operações SCRUD são realizadas sobre vistas do utilizador (Views) em formato de grelha de horários. Nesta altura, assim que o

primeiro método ficou 100% a funcionar, o risco de a aplicação não ficar pronta dentro do prazo caiu bastante, mesmo com o registo do atraso ao final do Sprint.

Tarefa Id 5 – Dia 75-90: 3º Sprint – módulo slots, Autorização e autenticação. Aqui correu conforme o planeado, inclusive foi possível eliminar os atrasos que se arrastavam através das tarefas anteriores. Aqui a aplicação estava pronta para ser colocada em produção.

Tarefa Id 6 – Dia 91-100: Escrita do relatório, ou seja, este documento. A maior dificuldade encontrada está relacionada com as pequenas diferenças de semântica que existem entre as culturas Portuguesa e Brasileira. Pois é comum o autor deste relatório se encontrar na seguinte situação quando escreve para o leitor perceber: quando deve escrever em poucas palavras, acaba por escrever muito e o contrário, ou seja, quando deve escrever mais, acaba por escrever pouco. Mas acredita-se que com o passar do tempo tem aprendido e melhorado.

Tarefa Id 7 – Dia 101-108: Esta tarefa ocorrerá depois da entrega deste documento, pelo que aqui se espera que pequenos ajustes, que não foram detectados nos testes de aceitação do final de cada Sprint, possam ser implementados (refinamentos) e a elaboração e comunicação do horário do segundo semestre de 2009 aconteça na nova aplicação.

4.3 Artefactos para entregar ao cliente

Para além da aplicação instalada e a funcionar em um computador disponibilizado pelo DEI, será entregue um compact disk (CD) com uma cópia dos seguintes artefactos:

- 1 Este documento – que contém: “Product backlog” (secção 8.1); Especificação dos testes (secção 8.2); e Guia de início rápido do utilizador (secção 8.5)
- 2 Código fonte – que contém o conjunto de ficheiros com o código da aplicação;

É do DEI o direito de fazer com este material o que julgar adequado. Uma sugestão seria submeter a aplicação à licença do tipo GNU/GPL 7, isto poderia atrair outros interessados no tema e estimular a troca de experiências.

4.4 Componentes

Para implementação da aplicação serão utilizados os componentes apresentados a seguir. A razão da escolha (pois poderiam ser outros componentes) se dá ao facto de que eles são suficientes (em relação a linguagem PHP 7, 7 e 7) para o propósito do projecto e devido à restrição do tempo (90 dias) são os componentes os quais a equipa de desenvolvimento está mais à vontade tentar atender as expectativas. Foi avaliado o risco de ao tentar implementar com outros componentes (por exemplo PostgreSQL 7, Glassfish 7 e JAVA Server Faces 7) o projecto poderia não ficar pronto no tempo disponível.

4.4.1 Servidor de base de dados

Na arquitectura da aplicação, este componente guardará os dados de forma persistente. Baseado no que foi dito até aqui, foi escolhido o MySQL versão 5.0 7. Como ferramentas para o design e a gestão da base de dados serão utilizados MySQL Workbench 7 e GUI Tools 7, respectivamente.

4.4.2 Servidor Web

Este componente provê o serviço de páginas Web (XHTML, CSS e PHP). Foi escolhido o Apache HTTPd versão 2.2 7.

Para melhorar o nível de segurança da comunicação entre os clientes e o servidor Web, será configurado no servidor os protocolos de comunicação com criptografia SSL/TLS através do kit de ferramentas OpenSSL 7.

4.4.3 Linguagem da aplicação

Este componente será utilizado para interpretar a lógica e manipular de forma dinâmica os conteúdos XHTML 7, CSS 7 e Javascript 7. Foi escolhido o PHP versão 5.2 7.

4.4.4 Framework PHP

Este componente será utilizado para ajudar no desenvolvimento da aplicação na linguagem PHP, utiliza “design patterns” Modelo, Vista e Controlador (MVC) 7. Em muitos casos o desenvolvimento utilizando Framework pode ajudar a implementar as aplicações com maior qualidade 7. Foi escolhido o CakePHP versão 1.2 7, mas poderia ter sido outra 7.

4.4.5 Integrated Development Environment (IDE)

Este componente será utilizado para codificar a aplicação. Procurou-se uma ferramenta que permitisse: (i) visão integrada de todo o design do projecto; (ii) realce de sintaxe e outros recursos de edição em PHP, XHTML, JavaScript e CSS; e (iii) um depurador de código. Foi escolhido o NETBEANS 6.5 com suporte a PHP 7.

4.4.6 Cliente da aplicação

Em princípio quaisquer dos actuais navegadores (browsers) 7 poderão ser utilizados pelos actores (Equipa de gestão, Professores, Alunos). Neste projecto, todos os testes foram realizados com o Mozilla Firefox versão 3 7.

Em relação à segurança na comunicação com o servidor, a maioria dos navegadores actuais já trazem por defeito os protocolos de comunicação com criptografia SSL/TLS 7.

Capítulo 5. Trabalho realizado

Este capítulo apresenta um breve comentário sobre os requisitos (“Product backlog”) e questões relacionadas ao nível de design, nomeadamente as classes, actores, casos de usos e base de dados, utilizados para perceber como a aplicação foi implementada.

5.1 Requisitos

Do que foi apresentado em 2.5.4, as entidades do negócio são os fundamentos para os módulos (secção 3.3) ao nível de **arquitectura** (secção 3.4) e para as **classes** (secção 5.2.1) ao nível de **design** (secção 5.2) da nova aplicação proposta para o DEI.

Os requisitos (ou histórias) do DEI ao nível de design estão sob a forma de “Product backlog” (secção 8.1).

5.1.1 Funcionais

O que foi apresentado até o momento (principalmente a secção 2.1 e o 3) ajuda a perceber quais os módulos devem ser implementados e as restrições que devem ser embutidas em tais módulos.

A investigação da arquitectura do SI actual (secção 2.5) também foi decisiva uma vez que ajudou a perceber a lógica e a solucionar (ou minimizar efeitos) dos pontos fracos ali existentes.

5.1.2 Não funcionais (qualidade)

Finalmente, o que foi apresentado, principalmente nas secções 2.2, 2.3, 2.4 e no 3, ajuda a perceber os aspectos de qualidade devem ser atendidos para o sucesso do projecto.

5.2 Design

Ao nível de design, cada classe será invocada a partir de chamadas na interface do utilizador que é orientada por um menu principal. As restrições, apresentadas na secção 3.2.2, e demais regras do negócio estarão embutidas nos métodos que manipularão os atributos destas classes. Os conteúdos dos atributos em sua maioria poderão ser guardados de forma persistente nos campos das tabelas na base de dados.

Ilustração 12 – Excerto da página inicial do sítio Web (Home)



5.2.1 Classes

5.2.1.1 Classes tipo

Do que foi dito para os módulos `tipo` (secção 3.3) ao nível de **arquitectura** vai aqui reflectir para as **classes**, ou seja, existe uma classe que representa o respectivo módulo tipo e esta terá métodos que manipulam os atributos de forma semelhante a outras classes de outros módulos tipo, daí elas serem denominadas `classes tipo`. Excepto Slot e Horário (Timetable) todas as classes são `classes tipo`.

Para as `classes tipo`, a aplicação deve possibilitar as operações para fazer pesquisa (Search), adicionar (Create), exibir (Read), alterar (Update) ou excluir (Delete) os dados que manipula. Pelo que denominamos operações SCRUD.

5.2.1.2 Outras classes

- 1 Slot – para além de fazer o que foi apresentado para a classe tipo, esta classe tem um método para copiar todos os slots de um período para outro numa só etapa.
- 2 Horário – em relação às `classes tipo` tem de igual apenas os métodos Adicionar e Excluir. Já o método de pesquisa é diferente, pelo que são diversos (contra apenas um que ocorre na classe tipo) os quais vão representar os relatórios.

5.2.2 Actores

São as entidades (pessoas ou não) que interagem no contexto da aplicação.

5.2.2.1 Equipa de gestão do horário

São os que utilizarão total ou parcialmente as funcionalidades da aplicação. Usualmente formada por funcionários do DEI. O principal caso de uso aqui será um ciclo completo da elaboração do horário a cada novo período.

5.2.2.2 Professores

São os utilizadores da aplicação que serão associados às aulas. Esta associação é realizada por algum membro da equipa de gestão do horário.

5.2.2.3 Público

São os alunos do DEI (ou qualquer outra pessoa) interessados em aceder a área pública (Home) para consulta das grelhas de horários. Não serão utilizadores com Username e Password, como acontece com os anteriores.

5.2.2.4 Aplicação de utilização do DEI

A própria aplicação pode ser considerada um actor, visto que haverá data interchange entre esta e as demais aplicações utilizadas pelo DEI, como por exemplo o WoC.

5.2.3 Casos de uso

5.2.3.1 Login na área restrita

Para aceder à área restrita, faz-se necessário ter credenciais (username & password) e conforme o grupo do utilizador a que pertencer, estará apto a realizar um determinado conjunto de acções na aplicação.

Aceder ao sítio Web – Escolher qualquer um dos módulos (Periods, Slots, Courses, ..., Timetables, excepto Home) – Acesso negado, fazer login na aplicação informando Username e Password – resposta: Se inválido, verificar o que está errado; senão, Fim.

5.2.3.2 Operações em classes tipo

Será apresentado à seguir o caso de uso para a classe tipo Período, mas para as demais classes tipo, segue o mesmo raciocínio:

Ilustração 13 – Gestão da classe tipo “Período”



Periods

Page 1 of 1, showing 8 records out of 8 total, starting on record 1, ending on 8

	Id	Name	Comments	Created	Modified
<input type="checkbox"/>	1	View Edit	2009-2	2009-06-22 13:54:47	2009-06-22 13:54:47
<input type="checkbox"/>	2	View Edit	2009-1	2009-06-22 13:54:47	2009-06-22 13:54:47
<input type="checkbox"/>	6	View Edit	2008-2	2009-06-22 13:54:47	2009-06-22 13:54:47
<input type="checkbox"/>	5	View Edit	2008-1	2009-06-22 13:54:47	2009-06-22 13:54:47
<input type="checkbox"/>	4	View Edit	2007-2	2009-06-22 13:54:47	2009-06-22 13:54:47
<input type="checkbox"/>	3	View Edit	2007-1	2009-06-22 13:54:47	2009-06-22 13:54:47
<input type="checkbox"/>	7	View Edit	2006-2	2009-06-25 12:27:07	2009-06-25 12:27:07
<input type="checkbox"/>	8	View Edit	2006-1 teste	2009-06-25 16:39:33	2009-06-25 16:39:33

<< previous | next >>

Search name

You're logged in as root  

- 1 Pesquisar: após login bem-sucedido e estando no módulo de período – Pesquisar/filtrar período desejado – Resposta: dados desejados ou vazio. Fim.
- 2 Adicionar: após login bem-sucedido e estando no módulo de período – adicionar novo período – preencher conteúdo dos campos – guardar o conteúdo – Se conteúdo inválido, pede para verificar o que está errado; senão, Fim.
- 3 Visualizar: após login bem-sucedido e estando no módulo de período – seleccionar período desejado para visualizar o conteúdo – Fim.
- 4 Alterar: após login bem-sucedido e estando no módulo de período – seleccionar período desejado para editar o conteúdo – alterar conteúdo dos

campos – guardar as alterações – Se conteúdo inválido, pede para verificar o que está errado; senão, Fim.

- 5 Excluir: após login bem-sucedido e estando no módulo de período – seleccionar período(s) desejado(s) – excluir período – Fim.

5.2.3.3 Operações em outras classes

- 1 Slot – para além de fazer o que foi apresentado na classe tipo, é possível copiar todos os slots de um período para outro numa só etapa. Após login bem-sucedido e estando no módulo de slot – Seleccionar período de origem (que possui os slots que se deseja copiar) – Seleccionar período de destino – executar a cópia – Fim.
- 2 Horário – as operações de Adicionar, Excluir, Pesquisar e Visualizar são realizados sob a forma de grelha de horários, daí a diferença na implementação dos métodos em relação as demais classes. Inclusive tais métodos vão representar alguns dos casos de uso dos relatórios.

Ilustração 14 – Método adicionar da classe horário

Add Timetable

Currentclass
2008-2 MEI EPMult T1

Slots for this Currentclass: Total 2, you have used 1, remained 1.

	MON	TUE
08:30-09:00	<input type="checkbox"/>	<input type="checkbox"/>
09:00-10:00	<input type="checkbox"/>	MEI EPMult (T1) B2
10:00-11:00	<input type="checkbox"/>	<input type="checkbox"/>
11:00-12:00	<input type="checkbox"/>	<input type="checkbox"/>
12:00-13:00	<input type="checkbox"/>	<input type="checkbox"/>
13:00-14:00	<input type="checkbox"/>	<input type="checkbox"/>
14:00-15:00	<input type="checkbox"/>	<input type="checkbox"/>
15:00-16:00	<input type="checkbox"/>	<input type="checkbox"/>
16:00-17:00	<input type="checkbox"/>	<input type="checkbox"/>
17:00-18:00	<input type="checkbox"/>	<input type="checkbox"/>
18:00-19:00	<input type="checkbox"/>	<input type="checkbox"/>
19:00-20:00	<input type="checkbox"/>	<input type="checkbox"/>

Comments

Submit Back Cancel Reset

[Add](#) [Delete](#) [List per Course](#) [List per Subject](#) [List per](#)

5.2.4 Casos de uso de relatórios

Alguns dos relatórios estão disponíveis para consulta na área pública (Home) e outros na área restrita do sítio Web.

5.2.4.1 Na área pública

- 1 Grelha de horários por curso (List per Course): aceder ao sítio Web – Escolher List per Course – Seleccionar período – Seleccionar curso – Visualizar relatório – Fim.

- 2 Outros relatórios na área pública – seguem o mesmo raciocínio do relatório de Horário por: (i) Disciplina (List per Subject); (ii) Sala (List per Room); e (iii) Utilizador (List per User).

Ilustração 15 – Relatório horário por curso

Period: 2008-2 Course: MEI

	MON	TUE	WED	THU	FRI
08:30-09:00					
09:00-10:00		EDJ T1 (B1) L Roque; IHC PL1 (C5.1) L Roque; EPMult T1 (B2) NO_User	EDJ PL1 (C5.1) L Roque; GPS T1 (B1) NO_User; IN T1 (E4.7) NO_User; IHC T1 (B2) L Roque	ER PL1 (C5.1) P Simões; GPS PL1 (C5.2) NO_User; IN PL1 (E4.5) NO_User	ER T1 (B1) P Simões; GE T1 (BP-Anf.A) NO_User
10:00-11:00		EDJ T1 (B1) L Roque; IHC PL1 (C5.1) L Roque	EDJ PL1 (C5.1) L Roque; GPS T1 (B1) NO_User; IN T1 (E4.7) NO_User; IHC T1 (B2) L Roque	ER PL1 (C5.1) P Simões; GPS PL1 (C5.2) NO_User; IN PL1 (E4.5) NO_User	ER T1 (B1) P Simões; GE T1 (BP-Anf.A) NO_User
11:00-12:00	API T1 (B1) L A Silva				GE PL1 (BP-Anf.A) NO_User
12:00-13:00	API T1 (B1) L A Silva				GE PL1 (BP-Anf.A) NO_User

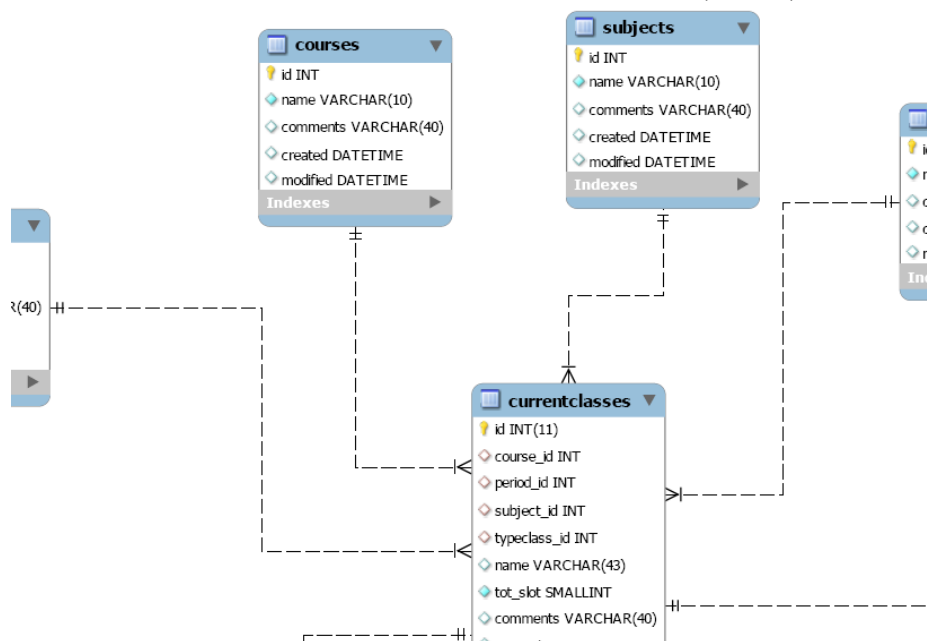
5.2.4.2 Na área restrita

É possível aceder aos mesmos relatórios da área pública e mais, o relatório de Conflito de utilizadores no horário (List User’s Overlaps per slot): após login bem-sucedido e estando no módulo de horário – seleccionar período desejado – visualizar relatório – Fim.

5.2.5 Base de dados

Uma vez que os conteúdos dos atributos das classes em sua maioria poderão ser guardados de forma persistente nos campos das tabelas na base de dados, segue alguns comentários sobre a implementação desta. O diagrama completo da base de dados pode ser visto na secção 8.3.

Ilustração 16 – Estrutura da base de dados (excerto)



5.2.5.1 Estrutura

Procurou-se adequar a estrutura da base de dados (secção 8.4) adoptando as convenções sugeridas pela da Framework cakePHP, justamente para tirar algumas vantagens do que ela oferece 7.

Para cada tabela há um campo auto incremento de nome ID. Também para cada tabela há um campo VARCHAR denominado NAME, visto que a Framework gera listas para popular as “SelectBoxes” automaticamente quando tais campos estão presentes, seria esta a primeira razão para retirar da forma normal 7 as tabelas.

A segunda razão seria o de evitar comandos SELECT para buscar os dados que estão nas tabelas relacionadas.

Finalmente, a terceira razão seria de que os campos NAME foram aproveitados para a utilização do método de pesquisa (search) dos dados nas tabelas. Se não utilizássemos o campo NAME, aqui teríamos mais SELECTS extras para fazer buscas nos dados das tabelas relacionadas quando o utilizador realizasse uma operação de pesquisa.

Assim para este projecto, acredita-se ser menos custoso ter esta não normalização com um conjunto de operações sobre o campo NAME do que realizar SELECTS extras.

Foi implementado na base de dados um conjunto de gatilhos, os chamados TRIGGERS, para manter actualizado o conteúdo do campo NAME das tabelas. À seguir um excerto de um dos TRIGGERS. Todos os triggers assim como a definição das tabelas podem ser consultadas na secção 8.4.

```
CREATE TRIGGER currentclass_name_ins BEFORE INSERT ON currentclasses
FOR EACH ROW BEGIN
  SELECT name INTO @v_name_course FROM courses WHERE id=NEW.course_id;
  SELECT name INTO @v_name_period FROM periods WHERE id=NEW.period_id;
  SELECT name INTO @v_name_subject FROM subjects WHERE id=NEW.subject_id;
  SELECT name INTO @v_name_typeclass FROM typeclasses WHERE id=NEW.typeclass_id;
  SET NEW.name= CONCAT(@v_name_period,' ',@v_name_course,' ',@v_name_subject,'
',@v_name_typeclass);
END;
```

5.2.5.2 Integridade dos dados

Para garantir a integridade referencial dos dados das tabelas procurou-se embutir, o quanto possível, as regras na base de dados. Assim foram utilizados como principais recursos, as chaves primárias e chaves forasteiras para relacionar as tabelas, chaves únicas, para manter a integridade do campo NAME.

Contudo, ainda foi necessário ter algumas regras de integridade fora da base de dados, ou seja, embutidas nos métodos das classes dos controladores da aplicação. Um exemplo seria a questão dos slots (available) disponíveis para uma aula (currentclass).

Tendo em conta que a relação entre as tabelas currentclasses e timetables é do tipo “um para muitos”, depois de informar um valor (digamos 2) para o campo tot_slot na tabela currentclasses, de seguida lançar os dois slots na tabela timetables. Para prevenir que o utilizador altere o valor do campo tot_slot em currentclasses e então ocorra a perda da integridade dos dados, foi implementado no método edit do controlador de currentclasses de forma que o campo tot_slot fica como somente leitura na visão (View) do utilizador.

Capítulo 6. Conclusões e perspectivas futuras

Este capítulo apresenta as principais conclusões sobre o que foi este estágio, alguns dos resultados esperados; e acções que poderão ser realizadas no futuro para estender ou melhorar o que foi feito até aqui.

6.1 Caracterização

Do que foi apresentado nos capítulos anteriores, acredita-se que para o DEI o objectivo foi alcançado de forma que se tem a possibilidade de elaborar e comunicar os horários académicos dos cursos que oferece com mais qualidade.

Para o orientando, também foram atendidos os objectivos, porque para além de consolidar os conhecimentos adquiridos ao longo do Mestrado, acredita que fez as “escolhas correctas” quando assumiu o compromisso em torno deste tema, da forma como se dedicou, acabando por entregar a aplicação a funcionar no espaço de tempo disponível.

6.2 Resultados esperados

Espera-se que a forma actual como o DEI elabora e comunica seus horários (secção 2.5.2) tenha sido melhorada, visto que a nova aplicação consegue o mesmo resultado eliminando as etapas 2 e 3 e para a etapa 4 (secção 2.5.2) pretende que a troca de dados aconteça de forma a sofrer a menor intervenção manual que for possível.

E mais, que as fraquezas observadas no SI actual (secção 2.5.3) tenham sido parcialmente ou plenamente sanadas.

6.3 Lições aprendidas

- 1 Em um espaço de noventa dias para implementar uma aplicação, deve-se perceber o problema, reunir tudo o que foi aprendido e necessário para buscar as soluções e iniciar o trabalho. O espaço para novas experiências em cenários como este projecto é mínimo ou inexistente.
- 2 Mesmo que inicialmente penalizado, deve-se tentar mudar de direcção quando estiver em risco todo o percurso que foi construído até chegar aqui;
- 3 Se for facultada a escolha, deve ficar com os componentes que está mais ambientado, pois no final para os avaliadores do trabalho (utilizadores e Júri) poderá ter importância secundária o conjunto de componentes que foi adoptado;
- 4 Os componentes utilizados para implementar a solução (secção 4.4) se não são os melhores, no mínimo estão próximos disto, pois milhares de pessoas, empresas e comunidades espalhadas ao redor do planeta os utilizam;

- 5 Desenvolvimento iterativo e incremental SEMPRE, que o cenário for semelhante a este projecto. Tentar aqui com ciclo de vida em Cascata (waterfall) por exemplo, aumenta muito as possibilidades de insucesso do projecto.
- 6 Levantar requisitos e envolver o cliente no projecto representam uma grande dificuldade, não é de se estranhar a farta literatura em torno destes temas.
- 7 Framework! No início é requerido um esforço para se adaptar ao standard que impõem, mas de seguida os resultados são compensadores.
- 8 DBMS! Pode representar uma dificuldade inicial ter de modelar a base de dados à lógica do negócio, mas vencida esta dificuldade o resultado é compensador.

6.4 Trabalho futuro

Apesar do que foi realizado até o momento, há ainda uma série de melhorias e novas funcionalidades que podem melhorar muito a aplicação.

Ao nível de design:

- 1 Visto ser uma restrição facultativa (secção 3.2.2, SC 3.2.2, página 10) e o prazo disponível de entrega da aplicação, o módulo slots do utilizador (slot_user) não foi implementado. No futuro poderá ser interessante ter esta funcionalidade para que os professores possam lançar suas preferências de horários para ministrar as aulas e a aplicação possa apresentar os slots disponíveis no momento da elaboração da grelha, de acordo com mais este critério.
- 2 Recursos de arrastar e soltar (drag 'n drop) os slots de disciplinas ocupados na grelha de horários para melhorar a usabilidade em relação à elaboração do horário 7;
- 3 Estender o método de cópia dos registos de slots de um período para outro, para permitir também que sejam copiados em uma só acção do utilizador as aulas (currentclass) e os professores associados às aulas (currentclass_user);
- 4 Definir um standard de ficheiro texto plano ou XML com a equipa das demais aplicações de utilização do DEI para gerar um módulo que em uma única acção do utilizador gere todos os ficheiros de exportação necessários ao intercâmbio de dados;
- 5 Melhorar o tratamento das mensagens de erro na validação dos dados 7;
- 6 Suporte a internacionalização i18n 7 A interface do utilizador está na língua inglesa. Pode ser interessante implementar i18n, visto ser um país de língua portuguesa e receber estudantes de língua espanhola;

Ao nível de arquitectura, seria conduzir experiências para verificar os limites de desempenho e escalabilidade: do sítio Web 7, 7; da base de dados 7, 7 da Framework 7 e do servidor Web 7 para implementar medidas que visem evitar possíveis estrangulamentos (bottlenecks) nestes componentes.

Capítulo 7. Referências

Este capítulo apresenta as referências utilizadas para realização deste estágio. Todas as referências foram consultadas no período de 02 de Abril a 10 de Julho de 2009.

- [1] MirHassani, S.A. *Improving paper spread in examination timetables using integer programming*. Applied Mathematics and Computation, Volume 179, Issue 2, 15 August 2006, Pages 702-706. Published by Elsevier Inc.
- [2] Google Directory. *Scheduling Utilities*. http://directory.google.com/Top/Computers/Software/Educational/Administration_and_School_Management/Scheduling_Uilities/
- [3] aSc Timetables. *The aSc Timetables Software*. <http://www.asctimetables.com/>
- [4] Cyon Canada Inc. *About Scientia*. <http://www.scientia.com/ca/aboutscientia/>
- [5] The Shuttleworth Foundation. *The SchoolTool Book*. http://book.schooltool.org/htmlhelp/what-is-1_0.html
- [6] Richardson, L. *Open Administration for Schools 3.50 User Documentation*. <http://richtech.ca/openadmin/docs/userdoc/index.html>
- [7] [TimeFinder Project. *User Documentation*. <http://timefinder.de/>
- [8] intime. *Interactive School Timetabling*. <http://intime.sourceforge.net/>
- [9] Flor de Utopia. *Web On Campus*. <https://woc.dei.uc.pt/weboncampus/>
- [10] UniTime. *University Timetabling – Comprehensive Academic Timetabling Solutions*. <http://www.unitime.org/>
- [11] UniTime. *System Highlights*. http://www.unitime.org/unitime_intro.php
- [12] Sun Microsystems. *Java EE Overview*. <http://java.sun.com/javaee/>
- [13] GNU. *General Public License*. <http://www.gnu.org/licenses/gpl.html>
- [14] UniTime. *Timetabling Installation*. http://wiki.unitime.org/Timetabling_Installation
- [15] UniTime. *Course Timetabling*. http://wiki.unitime.org/Timetabling_Documentation#Course_Timetabling
- [16] Unitime. *Presentations*. <http://www.unitime.org/presentations.php>
- [17] Unitime. *XML Interfaces*. http://www.unitime.org/uct_interfaces.php
- [18] Lalescu, L. *Open source free timetabling software for automatically scheduling the timetable of a school, high-school or university*. <http://lalescu.ro/liviu/fet/>
- [19] Nokia Corp. *Qt - A cross-platform application and UI framework*. <http://www.qtsoftware.com/>
- [20] Dirr, V. *FET Manual – Version 0.7*. <http://www.timetabling.de/manual/FET-manual.en.html>
- [21] FET. *Features*. <http://lalescu.ro/liviu/fet/features.html>
- [22] Yahoo groups. *FET - free timetabling software*. <http://tech.groups.yahoo.com/group/fet-timetabling/>
- [23] UniTime. *Demo Session*. http://www.unitime.org/uct_demo.php
- [24] Chant, C. *How to Convert HTML to Plain Text*. http://www.ehow.com/how_5040568_convert-html-plain-text.html
- [25] Add ons for Firefox. *PDF download*. <https://addons.mozilla.org/en-US/firefox/addon/636>

- [26] Sommerville, I. *Software Engineering, 7th edition. Chapter 1 ed. 2004.* <http://www.cs.st-andrews.ac.uk/~ifs/Books/SE7/Presentations/PDF/ch1.pdf>
- [27] Pressman, R. S. *Engenharia de Software.* 3ª ed. 30-45 p. Makron Books (1995).
- [28] Kniberg, H. *Scrum and XP from the Trenches.* <http://www.infoq.com/minibooks/scrum-xp-from-the-trenches/>
- [29] PostgreSQL Group. *Documentation.* <http://www.postgresql.org/docs/>
- [30] GlassFish Project. *Documentation Home Page.* <https://glassfish.dev.java.net/docs/project.html>
- [31] Sun Microsystems. *JavaServer Faces Technology.* <http://java.sun.com/javaee/javaserverfaces/>
- [32] Syllogistic Software Inc. *Web Development: A Comparison of Three Major Platforms.* http://www.syllogisticsoftware.com/papers/Web_Development_Technology_Comparison.html
- [33] Synodinos, D. G. *Is PHP Ready for the Enterprise?* <http://www.infoq.com/articles/enterprise-php>
- [34] Lemos, M. *How large is the PHP market?* <http://www.phpclasses.org/blog/post/95-How-large-is-the-PHP-market.html>
- [35] Sun Microsystems. *MySQL 5.0 Reference Manual.* <http://dev.mysql.com/doc/refman/5.0/en/index.html>
- [36] Sun Microsystems. *MySQL Workbench Community OSS Edition.* <http://dev.mysql.com/workbench/>
- [37] Sun Microsystems. *MySQL GUI Tools.* <http://dev.mysql.com/downloads/gui-tools/5.0.html>
- [38] Apache Group. *HTTPd Server Project.* <http://httpd.apache.org/>
- [39] OpenSSL Group. *Welcome to the OpenSSL Project.* <http://www.openssl.org/>
- [40] W3C. *XHTML documentation.* <http://www.w3.org/TR/xhtml1/>
- [41] W3C. *Cascading Style Sheets documentation.* <http://www.w3.org/Style/CSS/>
- [42] Mozilla Developer Center. *JavaScript documentation.* <https://developer.mozilla.org/en/JavaScript>
- [43] PHP. *PHP documentation.* <http://www.php.net/docs.php>
- [44] CakePHP Project. *Understanding Model-View-Controller.* <http://book.cakephp.org/view/10/Understanding-Model-View-Controller>
- [45] Sharp, J. *Why you should be using a framework.* http://www.joshsharp.com.au/blog/view/why_you_should_be_using_a_framework
- [46] CakePHP Project. *Manual CakePHP framework.* <http://book.cakephp.org>
- [47] O'Brien, D. *PHP frameworks, Part 1: Getting started with three popular frameworks: Zend, symfony, CakePHP.* <http://www.ibm.com/developerworks/opensource/library/os-php-fwk1/>
- [48] NetBeans. *PHP Learning Trail.* <http://www.netbeans.org/kb/trails/php.html>
- [49] W3Schools. *What is the trend in browser usage?* http://www.w3schools.com/browsers/browsers_stats.asp
- [50] Mozilla. *Firefox 3.* <http://www.mozilla.com>
- [51] Kassner, M. *SSL/TLS Certificates: What you need to know.* <http://blogs.techrepublic.com.com/networking/?p=640>
- [52] CakePHP. *Model and Database Conventions.* <http://book.cakephp.org/view/24/Model-and-Database-Conventions>
- [53] Hillyer, M. *An Introduction to Database Normalization.* <http://dev.mysql.com/tech-resources/articles/intro-to-normalization.html>

- [54] Dieter. *Drag 'n drop tutorial with the CakePHP 1.2 & Ajax*.
http://dieter.plaetinck.be/drag_n_drop_tutorial_with_cakephp_ajax_prototype_scriptaculous
- [55] CakePHP. *Data Validation*. <http://book.cakephp.org/view/125/Data-Validation>
- [56] cakePHP. *Internationalization & Localization*.
<http://book.cakephp.org/view/161/Internationalization-Localization>
- [57] Souders, S. *High Performance Web Sites*. <http://stevesouders.com/hpws/>
- [58] Henderson, C. *Building Scalable Web Sites*. <http://www.oreillynet.com/pub/au/2613>
- [59] Donnison, A. *Beginner's Guide to Website Performance with MySQL and memcached*.
<http://www.mysqlconf.com/mysql2009/public/schedule/detail/7009>
- [60] Sun Microsystems. *Chapter 7 MySQL Optimization*.
<http://dev.mysql.com/doc/refman/5.0/en/optimization.html>
- [61] Curry, M. *Eight Ways to Speed Up the Performance of CakePHP Apps*.
<http://www.pseudocoder.com/archives/2009/03/17/8-ways-to-speed-up-cakephp-apps/>
- [62] Apache Group. *Apache Performance Tuning*. <http://httpd.apache.org/docs/2.2/en/misc/perf-tuning.html>

Capítulo 8. Anexos

Este capítulo apresenta as especificações dos requisitos do cliente na forma de “Product backlog”; o documento de especificação de testes da aplicação; o diagrama da base de dados; e finalmente, o manual do utilizador.

8.1 Product Backlog

8.1.1 Requisitos funcionais

Id(*1)	Nome	Notas	Dificuldades	% tempo projecto
1	Conhecer o sistema actual e regras do negócio	Ter uma visão da lógica de negócio do sistema actual e avaliar o que é possível aproveitar.	Não usa base de dados, A aplicação JSF é apenas parte do processo, ainda há manipulação dos ficheiros gerados e macros microsoft Excel. Da forma como está, o risco de não entregar no prazo é muito alto.	2
1a	Regras negócio e nova aplicação	Depois de conhecer as regras do negócio, o risco de implementar de raiz com componentes conhecidos parece menor do que continuar o que está pronto.	Implementar sem framework parece ser mais fácil, porém a qualidade do código ao final poderia dificultar a manutenção.	10
2	Arquitectura & design	Utilizar componentes conhecidos para construir a arquitectura. Elaborar o design seguindo as recomendações da framework.	Levantar requisitos com o Product Owner.	10
2.1	Nova framework PHP	Aprender usar a framework e mais rápido possível começar a implementar o sistema.	A documentação oficial nem sempre atende. É necessário buscar exemplos em foruns.	15
2.2	modelar base de dados	Modelagem da base de dados, ajustar para usufruir das 'facilities' que a framework provê.	Toda normalizada expepto campos Name, que aproveita recursos da framework. Triggers disparados pela base de dados para actualizar tais campos.	10
3	Módulo das classes tipo	Implementar operações CRUD sobre os módulos: Periods, Courses, Subjects, Typeclasses, Currentclasses, Group_Users, Users, Currentclass_Users, Rooms, Slot_Users	Só existiram até a implementação do primeiro módulo, depois foi rápido.	5
4	Módulo Horário	Implementar operações CRUD sobre o módulo: Timetables	Dificuldade em implementar a lógica das pesquisas sob a forma de grelhas de horário.	18
5	Módulo slots	Implementar operações CRUD sobre o módulo: Slots	Dificuldade em implementar a rotina de cópia de slots de um período para o outro.	5
5.1	Autorização e autenticação	Implementar login do utilizador na área restrita do sítio Web. Definir política de permissões pelo grupo de utilizadores (o que cada grupo pode fazer na aplicação)	O tutorial da documentação oficial deixou à desejar, foi utilizado um tutorial disponível no site da IBM(*2).	2
5.2	Pesquisa (Search)	Implementar operações de pesquisa sobre os módulos das Classes tipo e Slots	Mais uma vez a comunidade ajudou!	7
5.3	Relatórios	Implementar relatórios: List per Course; List per Subject; List per Room; List per User	Mais uma vez a comunidade ajudou!	6

(*1) - Corresponde às tarefas listadas no Diagrama de Gantt

(*2) - <http://www.ibm.com/developerworks/opensource/tutorials/os-php-cake1/index.html>

8.1.2 Requisitos não funcionais (qualidade)

Id	Nome	Notas	Dificuldades	% tempo projecto
q.1	Melhorar formulários	Validação dos campos; campos readonly; deixar os forms mais adequados à lógica de negócio.	Colocar um form com formato de grelha;	4
q.2	Definir navegação	Definir a lógica de navegação no sistema.		1
q.3	Cores para timetable	Implementar cores para sinalização no módulo timetable: Verde - slot vazio; Amarelo - slot ocupado com 1 recurso; Vermelho - slot ocupado com mais de 1 recurso; branco - não há slot	Em excepção do relatório LIST PER USER, Slot ocupado com mais de um recurso (cor vermelha) não indica conflito.	2
q.4	Ajustar CSS e layout da aplicação	Foram adequados à aplicação apartir do template da framework.		2
q.5	Relatório List User's Overlaps slot	Implementar relatório List User's Overlaps slot, na área restrita.		1

8.2 Especificação dos testes

Neste projecto foi necessário descrever e planear um conjunto de testes de forma a detectar erros ao implementar os módulos da aplicação a desenvolver. Estes testes e a sua execução possibilitam a verificação dos vários módulos da aplicação, possibilitando ainda uma métrica da qualidade da mesma. Assim as principais funcionalidades e interacções que a aplicação possui serão testadas.

A metodologia a aplicar irá consistir na verificação de todas as opções disponíveis para cada campo dentro de cada módulo, pois só assim haverá a certeza que todo o software implementado está devidamente a funcionar. Os testes serão exaustivos e deverão ser seguidos de uma forma rigorosa.

De referir ainda um ponto importante: os testes aqui apresentados foram orientados por funcionalidade e o facto de existir aprovação em todos os testes definidos garante que o software está de acordo com os requisitos definidos no “Product backlog”, sendo consequentemente satisfeitos os testes de aceitação (validação pelo “Product Owner”).

8.2.1 Descrição da tabela de testes

Nº do teste: A identificação atribuída a cada teste a efectuar. O nº do teste pode ter um ou dois algarismos, conforme a sequência de testes, seguidos de um underscore _ e uma nomenclatura de forma a identificar a localização do teste na aplicação.

Condições de execução: Descreve a localização necessária na aplicação aquando da realização do teste

Entradas: Dados submetidos no(s) campo(s) a testar.

Saídas esperadas: Descreve o comportamento que a aplicação poderá devolver. No caso de o teste gerar um erro, este deverá ser corrigido tendo em conta a mensagem que for obtida da aplicação.

Result: Não é mais do que o resultado da indicação se a aplicação se comportou ou não conforme o previsto tendo em conta o teste efectuado. O Resultado obtido indicará que o teste revelou o esperado (Ok); não esperado e que representa um bug na aplicação (Error); ou um alerta (Warn) ou seja o resultado foi diferente do esperado, mas não impedirá a utilização da funcionalidade.

8.2.2 Nomenclatura

P – página da área pública.

R – página de acesso à área restrita, ou seja, página de login do utilizador.

RP – página da área restrita, página inicial do módulo de períodos.

RPA – página da área restrita, página add do módulo de períodos.

RPE – página da área restrita, página edit do módulo de períodos.

RPV – página da área restrita, página view do módulo de períodos.

RPD – página da área restrita, página delete do módulo de períodos.

RCCA – página da área restrita, página add do módulo de aulas (currentclasses).

RS – página da área restrita, página inicial do módulo de slots.

RSCBP – página da área restrita, página Copy Between Periods do módulo de slots.

RTTA – página da área restrita, página add do módulo de horários (timetables).

RTTD – página da área restrita, página delete do módulo de horários (timetables).

RTTLUOS – página da área restrita, página List User's Overlaps per Slot do módulo de horários (timetables).

8.2.3 Tabela de testes

8.2.3.1 Relatório List per Course (área pública)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
P_1	Página inicial (home) do sítio Web. Clicar no link do nome do relatório.	Clicar no botão Next	Apresenta a grelha de horários com dados referente aos conteúdos das SelectBoxes ou vazio.	Ok
P_2	Página inicial (home) do sítio Web. Clicar no link do nome do relatório.	Escolher Period, escolher Course, clicar no botão Next	Apresenta a grelha de horários com dados referente aos conteúdos das SelectBoxes ou vazio.	Ok
P_3	Página inicial (home) do sítio Web. Clicar no link do nome do relatório.	Clicar no botão Close	Fecha a janela do relatório e volta para a página home.	Ok

Os demais relatórios (List per Subject, List per Room e List per User) da área pública seguem o mesmo raciocínio.

8.2.3.2 Login na área restrita

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
R_1	Página inicial (home) do sítio Web. Clicar no link de qualquer uma das opções (excepto Home) do menu, a aplicação irá direccionar o utilizador para a página de login.	Preencher os campos Username, com root, e password com xyz123, ou qualquer outro conteúdo que não faz match com o que está registado na aplicação. Clicar no botão Login	Apresenta a mensagem: Login failed. Invalid username or password. E aguarda por acção do utilizador.	Ok
R_2	Página inicial (home) do sítio Web. Clicar no link de qualquer uma das opções (excepto Home) do menu, a aplicação irá direccionar o utilizador para a página de login.	Preencher os campos Username, e password com conteúdo que faz match com o que está registado na aplicação. Clicar no botão Login	A aplicação verifica se o utilizador tem permissão para aceder a área restrita pretendida (opção do menu que fez o clique) e em caso afirmativo direcciona o utilizador para esta área. Em caso negativo exhibe a mensagem: You are not authorized to access that location.	Ok

8.2.3.3 Pesquisar (search) no módulo de períodos (Periods)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RP_1	Depois do login do utilizador, Clicar no link Period.	Colocar texto na caixa Search name, por exemplo -1, Clicar no botão Search	Apresenta na grelha localizada acima todos os registos que tem o conteúdo NAME LIKE "%-1%" informado.	Ok
RP_2	Depois do login do utilizador, Clicar no link Period.	Clicar no botão Search	Apresenta na grelha localizada acima todos os registos guardados na tabela.	Ok
RP_3	Depois do login do utilizador, Clicar no link Period.	Colocar texto na caixa Search name, por exemplo ABC, Clicar no botão Search	Apresenta na grelha localizada acima todos os registos que tem o conteúdo NAME LIKE "%ABC%" informado. Ou nenhum se o conteúdo informado não faz match com o campo NAME.	Ok

8.2.3.4 Adicionar um novo período (Periods)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RPA_1	Estando na página Period. Clicar no botão	Preencher campo Name com 2010-X, campo Comment com It's a	Apresenta as mensagens: The Period could not be saved. Please, try again e também Fill Year and Period, like	Ok

	Add.	comment, Clicar no botão Submit	yyyy-pp. For instance 2009-1. E aguarda por acção do utilizador.	
RPA_2	Estando na página Period. Clicar no botão Add.	Clicar no botão Submit	Apresenta as mensagens: The Period could not be saved. Please, try again e também Fill Year and Period, like yyyy-pp. For instance 2009-1. E aguarda por acção do utilizador.	Ok
RPA_3	Estando na página Period. Clicar no botão Add.	Clicar no botão Back	Retrocede uma acção do utilizador.	Ok
RPA_4	Estando na página Period. Clicar no botão Add.	Clicar no botão Cancel	Vai para a página inicial do módulo corrente, ou seja, vai para Period.	Ok
RPA_5	Estando na página Period. Clicar no botão Add.	Preencher campo Name com 2010-X, campo Comment com It's a comment, Clicar no botão Reset	Restabelece o conteúdo dos campos do formulário, ou seja, nesta opção por se tratar de um novo período, os campos voltam a ficar sem conteúdo.	Ok
RPA_6	Estando na página Period. Clicar no botão Add.	Preencher campo Name com 2008-2, campo Comment com It's a comment, Clicar no botão Submit	Apresenta as mensagens: The Period could not be saved. Please, try again e também This name has already been taken. E aguarda por acção do utilizador.	Ok
RPA_7	Estando na página Period. Clicar no botão Add.	Preencher campo Name com 2010-1, campo Comment com It's a comment, Clicar no botão Submit	Se não existe igual conteúdo do campo Name, guarda os dados e exhibe mensagem The Period has been saved.	Ok

8.2.3.5 Editar o conteúdo de um período (Periods)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RPE_1	Estando na página Period. Clicar no link Edit correspondente ao período que deseja alterar o conteúdo dos campos.	Alterar o conteúdo do campo Name com 2010-X ou 2008-2 ou com conteúdo vazio. O campo Comment com It's a comment, Clicar no botão Submit	Apresenta as mensagens: The Period could not be saved. Please, try again e também Fill Year and Period, like yyyy-pp. For instance 2009-1 ou This name has already been taken. E aguarda por acção do utilizador.	Ok
RPE_2	Estando na página Period. Clicar no link Edit correspondente ao período que deseja alterar o conteúdo dos campos.	Clicar no botão Back	Retrocede uma acção do utilizador.	Ok

RPE_3	Estando na página Period. Clicar no link Edit correspondente ao período que deseja alterar o conteúdo dos campos.	Clicar no botão Cancel	Vai para a página inicial do módulo corrente, ou seja, vai para Period.	Ok
RPE_4	Estando na página Period. Clicar no link Edit correspondente ao período que deseja alterar o conteúdo dos campos.	Preencher campo Name com 2010-X, campo Comment com It's a comment, Clicar no botão Reset	Restabelece o conteúdo dos campos do formulário, ou seja, nesta opção os campos voltam a exibir o conteúdo da última gravação.	Ok
RPA_7	Estando na página Period. Clicar no link Edit correspondente ao período que deseja alterar o conteúdo dos campos.	Preencher campo Name com 2010-2, campo Comment com It's a comment, Clicar no botão Submit	Se não existe igual conteúdo do campo Name, guarda os dados e exibe mensagem The Period has been saved.	Ok

8.2.3.6 Visualizar o conteúdo de um período (Periods)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RPV_1	Estando na página Period. Clicar no link View correspondente ao período que deseja visualizar o conteúdo dos campos.	Clicar no botão Back	Retrocede uma acção do utilizador.	Ok
RPV_2	Estando na página Period. Clicar no link View correspondente ao período que deseja visualizar o conteúdo dos campos.	Clicar no botão Edit	Vai para a secção 8.2.3.5.	Ok
RPV_3	Estando na página Period. Clicar no link View correspondente ao período que deseja visualizar o conteúdo dos campos.	Clicar no botão Add	Vai para a secção 8.2.3.4.	Ok
RPV_4	Estando na página Period. Clicar no link View correspondente ao período que deseja visualizar o conteúdo dos campos.	Clicar no botão Cancel	Vai para a página inicial do módulo corrente, ou seja, vai para Period.	Ok

8.2.3.7 Excluir um ou mais períodos (Periods)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RPD_1	Estando na página Period.	Seleccionar os períodos que deseja excluir, marcando a sua respectiva checkbox. Clicar no botão Delete.	A aplicação pede a base de dados que verifique a integridade referencial, neste caso se os Periods seleccionados para exclusão estão associados a alguma Currentclass, ou seja se há registos de periods na tabela de currentclasses. Em caso afirmativo, exibe a mensagem: Some Periods could not be deleted	Ok

			Periods. Não exclui os períodos seleccionados e volta para a página inicial do período. Em caso negativo, exclui os períodos seleccionados e exibe a mensagem: Periods deleted.	
RPD_2	Estando na página Period.	Não seleccionar qualquer período que deseja excluir. Clicar no botão Delete .	Apresenta a mensagem: No period has been selected. Está a exibir a mensagem: Some Periods could not be deleted.	Warn

8.2.3.8 Operações nos módulos Courses, Subjects, Typeclasses, Group_users, Users, Rooms e Slot_users

Seguem o mesmo raciocínio dos testes que foram apresentados até o momento. Assim:

Pesquisar – igual a secção 8.2.3.3.

Adicionar – igual a secção 8.2.3.4.

Editar – igual a secção 8.2.3.5.

Visualizar – igual a secção 8.2.3.6.

Excluir – igual a secção 8.2.3.7.

8.2.3.9 Adicionar uma nova aula (Currentclasses)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RCCA_1	Estando na página Currentclass. Clicar no botão Add.	Clicar no botão Submit	Apresenta a mensagem: You need to select at least one combination of Period, Course, Subject & Typeclass. Volta para a a página home de Currentclass e não está a exibir mensagem.	Warn
RCCA_2	Estando na página Currentclass. Clicar no botão Add.	Seleccionar um ou mais itens de cada uma das selectboxes. Clicar no botão Submit , deixando o campo tot_slot vazio.	Apresenta as mensagens: Currentclass could not be saved. Please, check it. e também Please enter a number between 1 and 1008. E aguarda por acção do utilizador.	Ok
RCCA_3	Estando na página Currentclass. Clicar no botão Add.	Seleccionar um ou mais itens de cada uma das selectboxes. Preencher campo tot_slot com 2, campo Comment com It's a comment, Clicar no botão Submit	Se não existe igual combinação de conteúdos dos campos period, course, subject & typeclass, guarda os dados e exibe mensagem The Currentclass has been saved. Se existe, exibe a mensagem: Currentclass could not be saved. Please, check it e The Period, Course, Subject and Typeclass is already taken, please choose another combination of values.	Ok

8.2.3.10 Demais operações em aulas (Currentclasses)

Pesquisar – igual a secção 8.2.3.3.

Editar – igual a secção 8.2.3.5.

Visualizar – igual a secção 8.2.3.6.

Excluir – igual a secção 8.2.3.7.

8.2.3.11 Cópia de slots entre períodos (Copy between periods)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RSCBP_1	Estando na página Slot. Clicar no botão Copy Between Periods .	Clicar no botão Submit	Apresenta a mensagem: Slots could not be saved. Please, check it. A aplicação não copia os slots. Era esperado a mensagem: You can not choose the same From and To periods.	Warn
RSCBP_2	Estando na página Slot. Clicar no botão Copy Between Periods .	No campo From period, seleccionar um período que não tem slots. No campo To period, seleccionar um período diferente do campo From Period Clicar no botão Submit	Apresenta a mensagem: Slots could not be saved. Please, check it. A aplicação não copia os slots. Era esperado a mensagem: You need choose From period with slots.	Warn
RSCBP_3	Estando na página Slot. Clicar no botão Copy Between Periods .	No campo From period, seleccionar um período que tem slots. No campo To period, seleccionar um período que já tem slots e seja diferente do campo From Period Clicar no botão Submit	Apresenta a mensagem: Slots could not be saved. Please, check it. A aplicação não copia os slots. Era esperado a mensagem: You need choose To period without slots.	Warn
RSCBP_4	Estando na página Slot. Clicar no botão Copy Between Periods .	No campo From period, seleccionar um período que tem slots. No campo To period, seleccionar um período que não tem slots e seja diferente do campo From Period Clicar no botão Submit	Guarda os dados e exhibe mensagem All Slots has been saved.	Ok

8.2.3.12 Demais operações em Slots

Pesquisar – igual a secção 8.2.3.3.

Adicionar – igual a secção 8.2.3.4.

Editar – igual a secção 8.2.3.5.

Visualizar – igual a secção 8.2.3.6.

Excluir – igual a secção 8.2.3.7.

8.2.3.13 Adicionar novos horários (Timetables)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RTTA_1	Estando na página Timetable. Clicar no botão Add.	Escolhe uma Currentclass, Clicar no botão Next.	Apresenta a mensagem: All idle rooms & slots for entire period are read to be displayed. Please click on 'Next' button. Utilizador deve clicar no botão Next para a próxima etapa.	Ok
RTTA_2	RTTA_1	Selecciona a quantidade de slots (marcando as respectivas checkboxes) em superior ao disponível (slots remained). Clica no botão Submit .	Apresenta a mensagem: Some Timetable could not be saved. Please, try again. E aguarda por acção do utilizador.	Ok
RTTA_3	RTTA_1	Não selecciona quaisquer slots (marcando as respectivas checkboxes). Clica no botão Submit .	Apresenta a mensagem: Some Timetable could not be saved. Please, try again. E aguarda por acção do utilizador. Era esperado mensagem: You need select at least one slot & room.	Warn
RTTA_4	RTTA_1	Selecciona a quantidade de slots (marcando as respectivas checkboxes) em inferior ou igual número ao disponível (slots remained). Clica no botão Submit .	Apresenta a mensagem: All Timetable has been saved. Guarda os dados e volta para a cena de lançamento de novo horário.	Ok

8.2.3.14 Excluir horários (Timetables) existentes

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RTTD_1	Estando na página Timetable. Clicar no botão Delete.	Escolhe uma Currentclass, Clicar no botão Next .	Apresenta a grelha de slots de timetables da currentclass escolhida:	Ok
RTTD_2	RTTD_1	Selecciona os slots (marcando as respectivas checkboxes) desejados. Clica no botão Delete .	Apresenta a mensagem: Timetable deleted. Exclui os slots de timetables seleccionados.	Ok
RTTD_3	RTTD_1	Clica no botão Delete .	Não apresenta a mensagem. Vai para o início da página Delete. Deveria apresentar a mensagem: You need select at least one slot.	Warn

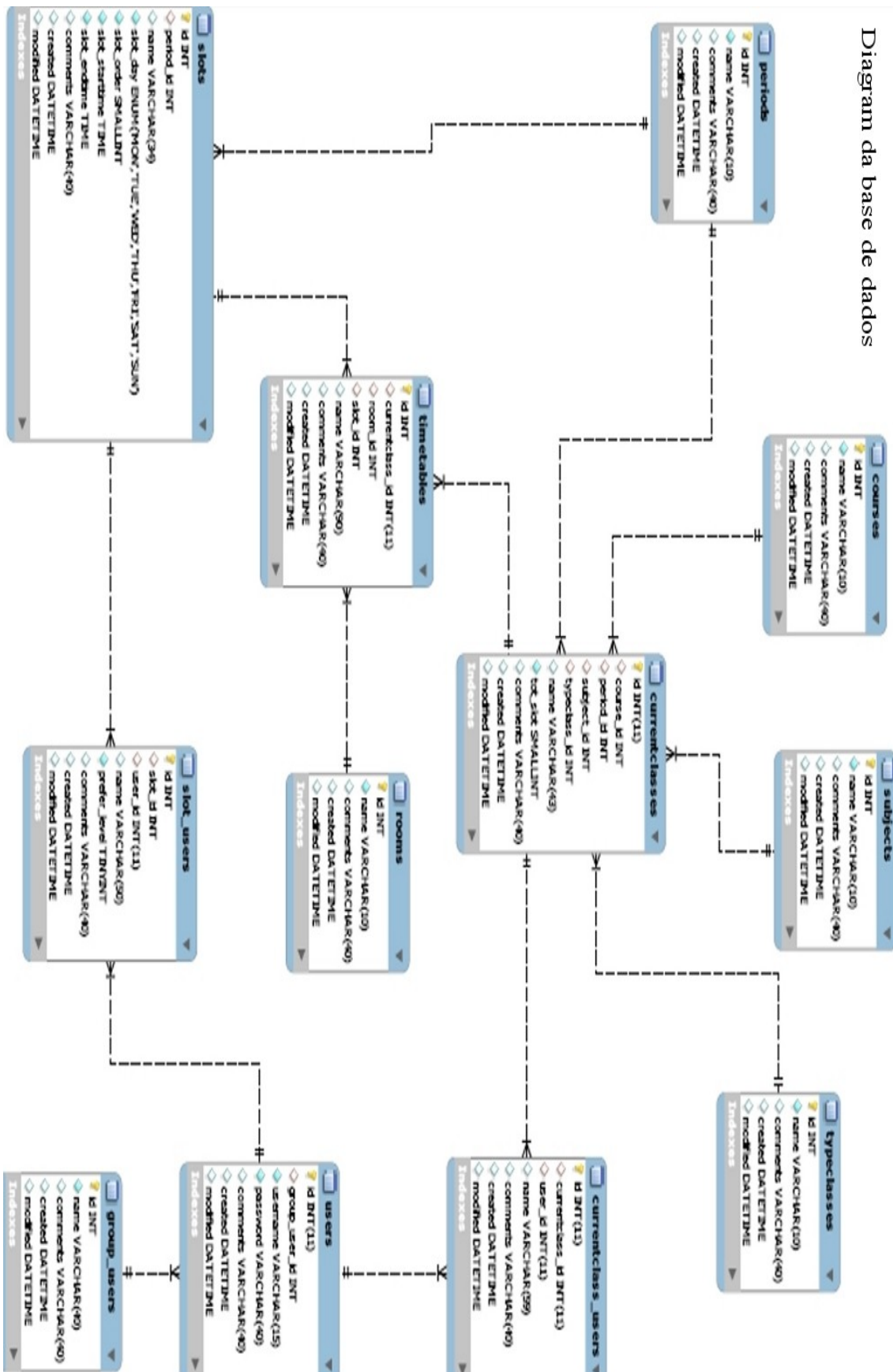
8.2.3.15 Relatório LUOS (List User's Overlaps per Slot)

Nº do teste	Condições de execução	Entradas	Saídas esperadas	Result
RTTLUOS_1	Estando na página Timetable. Clicar no link List User's Overlaps per Slot	Clicar no botão Next	Apresenta (se existir) a listagem de utilizadores que estão em mais de uma aula (timetables) no mesmo slot de horário para o período seleccionado.	Ok
RTTLUOS_1	Estando na página Timetable. Clicar no link List User's Overlaps per Slot	Escolhe um período. Clicar no botão Next	Apresenta (se existir) a listagem de utilizadores que estão em mais de uma aula (timetables) no mesmo slot de horário para o período seleccionado.	Ok

8.2.3.16 Demais relatórios do módulo de horários (Timetables)

Os relatórios List per Course, List per Subject, List per Room e List per User são os mesmos da área pública (secção 8.2.3.1).

8.3 Diagrama da base de dados



8.4 Definição da base de dados (script)

```

DROP DATABASE IF EXISTS `timetable`;

CREATE SCHEMA IF NOT EXISTS `timetable` DEFAULT CHARACTER SET
utf8 DEFAULT COLLATE utf8_general_ci;

USE `timetable`;

-----
-- Table `timetable`.`periods`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`periods` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(10) NOT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this model' ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `index1` (`name` ASC) );

-----
-- Table `timetable`.`slots`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`slots` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `period_id` INT NOT NULL ,
  `name` VARCHAR(34) NULL ,
  `slot_day` ENUM('MON','TUE','WED','THU','FRI','SAT','SUN')
NOT NULL DEFAULT 'MON' COMMENT 'Day of week. 1 = Mon, 2= Tue,
..., 7= Sun.' ,
  `slot_order` SMALLINT UNSIGNED NOT NULL COMMENT 'Slot
order' ,
  `slot_starttime` TIME NOT NULL COMMENT 'How many minutes has
one slot' ,
  `slot_endtime` TIME NOT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this table' ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL COMMENT 'concat
(Day,Slot_Number,Start_Time)' ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `index2` (`period_id` ASC, `slot_day` ASC,
`slot_order` ASC) ,
  INDEX `fk_period_ttables_periods` (`period_id` ASC) ,
  CONSTRAINT `fk_period_ttables_periods`
FOREIGN KEY (`period_id` )
REFERENCES `timetable`.`periods` (`id` )
ON DELETE NO ACTION
ON UPDATE NO ACTION);

-----
-- Table `timetable`.`courses`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`courses` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(10) NOT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this table' ,

```

```

`created` DATETIME NULL DEFAULT NULL ,
`modified` DATETIME NULL DEFAULT NULL ,
PRIMARY KEY (`id`) ,
UNIQUE INDEX (`name` ASC) );

-----
-- Table `timetable`.`subjects`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`subjects` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(10) NOT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this table' ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `index1` (`name` ASC) );

-----
-- Table `timetable`.`typeclasses`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`typeclasses` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(10) NOT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this table' ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `index1` (`name` ASC) )
COMMENT = 'type of classes';

-----
-- Table `timetable`.`currentclasses`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`currentclasses` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `course_id` INT NOT NULL ,
  `period_id` INT NOT NULL ,
  `subject_id` INT NOT NULL ,
  `typeclass_id` INT NOT NULL ,
  `name` VARCHAR(43) NULL DEFAULT NULL ,
  `tot_slot` SMALLINT UNSIGNED NOT NULL DEFAULT 1 COMMENT
'Total of slot on current day' ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this table' ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `fk_currentclasses_courses` (`course_id` ASC) ,
  INDEX `fk_currentclasses_periods` (`period_id` ASC) ,
  INDEX `fk_currentclasses_subjects` (`subject_id` ASC) ,
  INDEX `fk_currentclasses_typeclasses` (`typeclass_id` ASC) ,
  UNIQUE INDEX `index5` (`period_id` ASC, `course_id` ASC,
  `subject_id` ASC, `typeclass_id` ASC) ,
  CONSTRAINT `fk_currentclasses_courses`
    FOREIGN KEY (`course_id`)
    REFERENCES `timetable`.`courses` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_currentclasses_periods`
    FOREIGN KEY (`period_id`)
    REFERENCES `timetable`.`periods` (`id`)

```

```

        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
CONSTRAINT `fk_currentclasses_subjects`
    FOREIGN KEY (`subject_id` )
    REFERENCES `timetable`.`subjects` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_currentclasses_typeclasses`
    FOREIGN KEY (`typeclass_id`)
    REFERENCES `timetable`.`typeclasses` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
COMMENT = 'relation of course_period_subject_typeclasses';

-----
-- Table `timetable`.`group_users`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`group_users` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(40) NOT NULL ,
  `comments` VARCHAR(40) NULL ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `index1` (`name` ASC) )
ENGINE = InnoDB;

-----
-- Table `timetable`.`users`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`users` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `group_user_id` INT NOT NULL ,
  `username` VARCHAR(15) NOT NULL ,
  `password` VARCHAR(40) NOT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`) ,
  UNIQUE INDEX `index2` (`username` ASC) ,
  INDEX `fk_users_group_users` (`group_user_id` ASC) ,
  CONSTRAINT `fk_users_group_users`
    FOREIGN KEY (`group_user_id` )
    REFERENCES `timetable`.`group_users` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

-----
-- Table `timetable`.`currentclass_users`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`currentclass_users` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `currentclass_id` INT(11) NOT NULL ,
  `user_id` INT(11) NOT NULL ,
  `name` VARCHAR(59) NULL DEFAULT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this table' ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `fk_current_class_users_users` (`user_id` ASC) ,
  INDEX `fk_currentclass_users_currentclasses`
    (`currentclass_id` ASC) ,

```

```

UNIQUE INDEX `index3` (`currentclass_id` ASC, `user_id`
ASC) ,
CONSTRAINT `fk_current_class_users_users`
  FOREIGN KEY (`user_id` )
  REFERENCES `timetable`.`users` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_currentclass_users_currentclasses`
  FOREIGN KEY (`currentclass_id` )
  REFERENCES `timetable`.`currentclasses` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
COMMENT = 'course_period_subject_typeclass_users';

-----
-- Table `timetable`.`rooms`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`rooms` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(10) NOT NULL ,
  `comments` VARCHAR(40) NULL DEFAULT NULL COMMENT 'Comments of
this table' ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id` ) ,
  UNIQUE INDEX `index1` (`name` ASC) );

-----
-- Table `timetable`.`slot_users`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`slot_users` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `slot_id` INT NOT NULL ,
  `user_id` INT(11) NOT NULL ,
  `name` VARCHAR(50) NULL COMMENT
'concat(Username,Slot_Name)' ,
  `prefer_level` TINYINT UNSIGNED NOT NULL DEFAULT 1 COMMENT
'Slot preference level. 1 to first preference, 2 to second
and so on.' ,
  `comments` VARCHAR(40) NULL ,
  `created` DATETIME NULL DEFAULT NULL ,
  `modified` DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (`id` ) ,
  INDEX `fk_slot_users_slots` (`slot_id` ASC) ,
  INDEX `fk_slot_users_users` (`user_id` ASC) ,
  UNIQUE INDEX `index3` (`slot_id` ASC, `user_id` ASC) ,
  CONSTRAINT `fk_slot_users_slots`
    FOREIGN KEY (`slot_id` )
    REFERENCES `timetable`.`slots` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_slot_users_users`
    FOREIGN KEY (`user_id` )
    REFERENCES `timetable`.`users` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

-----
-- Table `timetable`.`timetables`
-----
CREATE TABLE IF NOT EXISTS `timetable`.`timetables` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `currentclass_id` INT(11) NOT NULL ,

```

```

`room_id` INT NOT NULL ,
`slot_id` INT NOT NULL ,
`name` VARCHAR(90) NULL ,
`comments` VARCHAR(40) NULL DEFAULT NULL ,
`created` DATETIME NULL DEFAULT NULL ,
`modified` DATETIME NULL DEFAULT NULL ,
PRIMARY KEY (`id`) ,
INDEX `fk_timetables_currentclasses` (`currentclass_id`
ASC) ,
INDEX `fk_timetables_rooms` (`room_id` ASC) ,
INDEX `fk_timetables_slots` (`slot_id` ASC) ,
UNIQUE INDEX `index4` (`slot_id` ASC, `room_id` ASC) ,
CONSTRAINT `fk_timetables_currentclasses`
  FOREIGN KEY (`currentclass_id` )
  REFERENCES `timetable`.`currentclasses` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_timetables_rooms`
  FOREIGN KEY (`room_id` )
  REFERENCES `timetable`.`rooms` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_timetables_slots`
  FOREIGN KEY (`slot_id` )
  REFERENCES `timetable`.`slots` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

# $Id: db_acl.sql 7945 2008-12-19 02:16:01Z gwoo $
#
# Copyright 2005-2008,      Cake Software Foundation, Inc.
#
# Licensed under The MIT License
# Redistributions of files must retain the above copyright
# notice.
# http://www.opensource.org/licenses/mit-license.php The MIT
# License

CREATE TABLE acos (
  id INTEGER(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  parent_id INTEGER(10) DEFAULT NULL,
  model VARCHAR(255) DEFAULT '',
  foreign_key INTEGER(10) UNSIGNED DEFAULT NULL,
  alias VARCHAR(255) DEFAULT '',
  lft INTEGER(10) DEFAULT NULL,
  rght INTEGER(10) DEFAULT NULL,
  PRIMARY KEY (id)
);

CREATE TABLE aros_acos (
  id INTEGER(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  aro_id INTEGER(10) UNSIGNED NOT NULL,
  aco_id INTEGER(10) UNSIGNED NOT NULL,
  _create CHAR(2) NOT NULL DEFAULT 0,
  _read CHAR(2) NOT NULL DEFAULT 0,
  _update CHAR(2) NOT NULL DEFAULT 0,
  _delete CHAR(2) NOT NULL DEFAULT 0,
  PRIMARY KEY(id)
);

CREATE TABLE aros (

```



```

    id INTEGER(10) UNSIGNED NOT NULL AUTO_INCREMENT,
    parent_id INTEGER(10) DEFAULT NULL,
    model VARCHAR(255) DEFAULT '',
    foreign_key INTEGER(10) UNSIGNED DEFAULT NULL,
    alias VARCHAR(255) DEFAULT '',
    lft INTEGER(10) DEFAULT NULL,
    rght INTEGER(10) DEFAULT NULL,
    PRIMARY KEY (id)
);

-- -----
-- Triggers
-- -----

#update name field for relation table before insert new row
DROP TRIGGER currentclass_name_ins;

DELIMITER ||

CREATE TRIGGER currentclass_name_ins BEFORE INSERT ON
    currentclasses
FOR EACH ROW BEGIN
    SELECT name INTO @v_name_course FROM courses WHERE
        id=NEW.course_id;
    SELECT name INTO @v_name_period FROM periods WHERE
        id=NEW.period_id;
    SELECT name INTO @v_name_subject FROM subjects WHERE
        id=NEW.subject_id;
    SELECT name INTO @v_name_typeclass FROM typeclasses WHERE
        id=NEW.typeclass_id;
    SET NEW.name= CONCAT(@v_name_period, ' ',@v_name_course, '
        ',@v_name_subject, ' ',@v_name_typeclass);
END;
||

DELIMITER ;

#update name field for relation table before update new row
DROP TRIGGER currentclass_name_upd;

DELIMITER ||

CREATE TRIGGER currentclass_name_upd BEFORE UPDATE ON
    currentclasses
FOR EACH ROW BEGIN
    IF (NEW.period_id!=OLD.period_id OR NEW.course_id!
        =OLD.course_id OR NEW.subject_id!=OLD.subject_id OR
        NEW.typeclass_id!=OLD.typeclass_id) THEN
    SELECT name INTO @v_name_course FROM courses WHERE
        id=NEW.course_id;
    SELECT name INTO @v_name_period FROM periods WHERE
        id=NEW.period_id;
    SELECT name INTO @v_name_subject FROM subjects WHERE
        id=NEW.subject_id;
    SELECT name INTO @v_name_typeclass FROM typeclasses WHERE
        id=NEW.typeclass_id;
    SET NEW.name= CONCAT(@v_name_period, ' ',@v_name_course, '
        ',@v_name_subject, ' ',@v_name_typeclass);
    END IF;
END;
||

DELIMITER ;

```

```

#update name field for relation table before insert new row
DROP TRIGGER currentclass_user_name_ins;

DELIMITER ||

CREATE TRIGGER currentclass_user_name_ins BEFORE INSERT ON
    currentclass_users
FOR EACH ROW BEGIN
    SELECT name INTO @v_name_currentclass FROM currentclasses
        WHERE id=NEW.currentclass_id;
    SELECT username INTO @v_name_user FROM users WHERE
        id=NEW.user_id;
    SET NEW.name= CONCAT(@v_name_currentclass, '
        ',@v_name_user);
    END;
||

DELIMITER ;

#update name field for relation table before update new row
DROP TRIGGER currentclass_user_name_upd;

DELIMITER ||

CREATE TRIGGER currentclass_user_name_upd BEFORE UPDATE ON
    currentclass_users
FOR EACH ROW BEGIN
    IF (NEW.currentclass_id!=OLD.currentclass_id OR
        NEW.user_id!=OLD.user_id) THEN
        SELECT name INTO @v_name_currentclass FROM currentclasses
            WHERE id=NEW.currentclass_id;
        SELECT username INTO @v_name_user FROM users WHERE
            id=NEW.user_id;
        SET NEW.name= CONCAT(@v_name_currentclass, '
            ',@v_name_user);
    END IF;
    END;
||

DELIMITER ;

#update name field for relation table before insert new row
DROP TRIGGER slot_name_ins;

DELIMITER ||

CREATE TRIGGER slot_name_ins BEFORE INSERT ON slots
FOR EACH ROW BEGIN
    SELECT name INTO @v_name_period FROM periods WHERE
        id=NEW.period_id;
    SET NEW.name= CONCAT(@v_name_period, ' ',NEW.slot_day, '
        ',NEW.slot_order, '
        ',SUBSTRING(NEW.slot_starttime,1,5));
    END;
||

DELIMITER ;

#update name field for relation table before update new row

```

```

DROP TRIGGER slot_name_upd;

DELIMITER ||

CREATE TRIGGER slot_name_upd BEFORE UPDATE ON slots
FOR EACH ROW BEGIN
    IF (NEW.slot_day!=OLD.slot_day OR NEW.slot_order!
        =OLD.slot_order OR NEW.slot_starttime!
        =OLD.slot_starttime OR NEW.period_id!=OLD.period_id)
    THEN
        SELECT name INTO @v_name_period FROM periods WHERE
            id=NEW.period_id;
        SET NEW.name= CONCAT(@v_name_period, ' ',NEW.slot_day, '
            ',NEW.slot_order, '
            ',SUBSTRING(NEW.slot_starttime,1,5));
    END IF;
END;
||

DELIMITER ;

#update name field for relation table before insert new row
DROP TRIGGER timetable_name_ins;

DELIMITER ||

CREATE TRIGGER timetable_name_ins BEFORE INSERT ON timetables
FOR EACH ROW BEGIN
    SELECT name INTO @v_name_room FROM rooms WHERE
        id=NEW.room_id;
    SELECT name INTO @v_name_slot FROM slots WHERE
        id=NEW.slot_id;
    SELECT name INTO @v_name_cclass FROM currentclasses WHERE
        id=NEW.currentclass_id;
    SET NEW.name= CONCAT(@v_name_cclass, ' ',@v_name_room, '
        ',@v_name_slot);
END;
||

DELIMITER ;

#update name field for relation table before update new row
DROP TRIGGER timetable_name_upd;

DELIMITER ||

CREATE TRIGGER timetable_name_upd BEFORE UPDATE ON timetables
FOR EACH ROW BEGIN
    IF (NEW.room_id!=OLD.room_id OR NEW.slot_id!=OLD.slot_id OR
        NEW.currentclass_id!=OLD.currentclass_id) THEN
        SELECT name INTO @v_name_room FROM rooms WHERE
            id=NEW.room_id;
        SELECT name INTO @v_name_slot FROM slots WHERE
            id=NEW.slot_id;
        SELECT name INTO @v_name_cclass FROM currentclasses WHERE
            id=NEW.currentclass_id;
        SET NEW.name= CONCAT(@v_name_cclass, ' ',@v_name_room, '
            ',@v_name_slot);
    END IF;
END;
||

```

```
DELIMITER ;

#update name field for relation table before insert new row
DROP TRIGGER slot_user_name_ins;

DELIMITER ||

CREATE TRIGGER slot_user_name_ins BEFORE INSERT ON slot_users
FOR EACH ROW BEGIN
    SELECT username INTO @v_name_user FROM users WHERE
        id=NEW.user_id;
    SELECT name INTO @v_name_slot FROM slots WHERE
        id=NEW.slot_id;
    SET NEW.name= CONCAT(@v_name_user, ' ',@v_name_slot);
END;
||

DELIMITER ;

#update name field for relation table before update new row
DROP TRIGGER slot_user_name_upd;

DELIMITER ||

CREATE TRIGGER slot_user_name_upd BEFORE UPDATE ON slot_users
FOR EACH ROW BEGIN
    IF (NEW.user_id!=OLD.user_id OR NEW.slot_id!=OLD.slot_id)
    THEN
        SELECT username INTO @v_name_user FROM users WHERE
            id=NEW.user_id;
        SELECT name INTO @v_name_slot FROM slots WHERE
            id=NEW.slot_id;
        SET NEW.name= CONCAT(@v_name_user, ' ',@v_name_slot);
    END IF;
END;
||

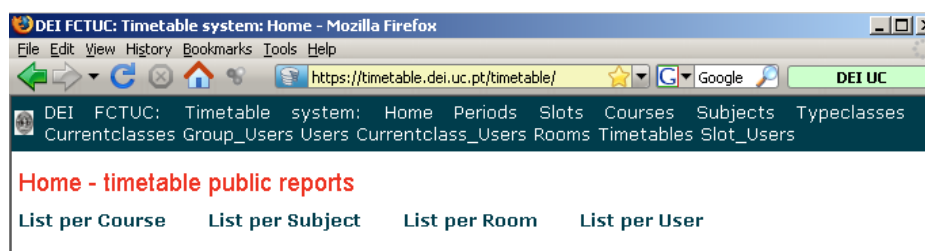
DELIMITER ;
```

8.5 Guia de início rápido do utilizador

8.5.1 Como aceder a aplicação

Inicie o seu navegador (browser) preferido e digite o endereço do sítio web <https://timetable.dei.uc.pt/timetable/>, a aplicação irá direccioná-lo para a área pública (Home).

Ilustração 17 – Página inicial da aplicação Web



8.5.2 Como gerar os relatórios na área pública

Existem 4 relatórios disponíveis na área pública:

List per Course – exibirá a grelha de horários por período e curso seleccionados.

List per Subject – exibirá a grelha de horários por período e disciplina seleccionados.

List per Room – exibirá a grelha de horários por período e sala seleccionados.

List per User – exibirá a grelha de horários por período e utilizador (do tipo professor) seleccionados.

1. Click no link **List per Course**. A aplicação abrirá uma segunda janela de nome **Timetables per Course**.
2. Informe o **Period** e o **Course** desejado.
3. Click no botão **Next**. O relatório será exibido.
4. Para encerrar o relatório pode fechar a janela clicando no emblema azul do Departamento de Engenharia Informática (DEI). A aplicação retorna o foco para a página Home do sítio Web.


Ilustração 18 – Relatório de horários por curso

	MON	TUE	WED	
08:30-09:00				
09:00-10:00		EDJ T1 (B1) L Roque; IHC PL1 (C5.1) L Roque; EPMult T1 (B2) NO_User	EDJ PL1 (C5.1) L Roque; GPS T1 (B1) NO_User; IN T1 (E4.7) NO_User; IHC T1 (B2) L Roque	ER PL1 (C5.2) L Roque
10:00-11:00		EDJ T1 (B1) L Roque; IHC PL1 (C5.1) L Roque	EDJ PL1 (C5.1) L Roque; GPS T1 (B1) NO_User; IN T1 (E4.7) NO_User; IHC T1 (B2) L Roque	ER PL1 (C5.2) L Roque
11:00-12:00		EPMult PL1 (C5.1) NO_User		

Para os restantes relatórios o raciocínio é análogo!

8.5.3 Como ligar e desligar da área restrita

Antes de tentar aceder a área restrita denotar que só será possível se estiver registado na aplicação com `username` e `password` e que pertença a um dos seguintes grupos de utilizadores: `Administrators` – terão permissões totais sobre a aplicação; `Managers` – poderão “montar as aulas” e elaborar a grelha de horários; e `Professors` – tem permissões para adicionar sua preferência de slots (esta funcionalidade não foi implementada). Para os demais objectos e recursos poderão apenas visualizar (mas não alterar).

1. Click no link de qualquer (excepto `Home`) um dos módulos do menu, por exemplo `Period`. A aplicação o direccionará para a janela de `login`.
2. Informe `Username` e `password`.
3. Click no botão `Login`. Se tiver informado os dados correctamente a aplicação o direccionará para a opção de menu anteriormente escolhida, no caso `Period`.
4. Para fazer `logout` (desligar-se do sistema com a conta de utilizador que acabou de ligar-se) basta fazer click no ícone  que aparece ao lado direito da mensagem `You're logged in as yourname`.

8.5.4 Como fazer pesquisa

Na área restrita da aplicação é possível fazer pesquisa e exibir apenas os dados desejados num determinado módulo, por exemplo, digamos que desejamos ver apenas alguns (e não todos) períodos no módulo de períodos:

1. Click no link `Period` do menu. A aplicação o direccionará para o módulo de períodos.
2. Na caixa `Search Name`, informe o conteúdo no qual deseja que a aplicação faça a recolha dos dados.
3. Click no botão `Search`. A aplicação exibirá (se houver) os dados com base no conteúdo informado.

Ilustração 19 – Pesquisa (search) de períodos

Periods

Page 1 of 1, showing 9 records out of 9 :

<input type="checkbox"/>	Id	Name
<input type="checkbox"/>	1 View Edit	2009-2
<input type="checkbox"/>	2 View Edit	2009-1
<input type="checkbox"/>	6 View Edit	2008-2
<input type="checkbox"/>	5 View Edit	2008-1
<input type="checkbox"/>	4 View Edit	2007-2
<input type="checkbox"/>	3 View Edit	2007-1
<input type="checkbox"/>	7 View Edit	2006-2
<input type="checkbox"/>	8 View Edit	2006-1
<input type="checkbox"/>	9 View Edit	2003-2

<< previous | next >>

Add Delete

Search name

Put here text of the field N

Search

8.5.4.1 Opções de pesquisa

Denotar que a pesquisa é feita com base no critério de busca SQL LIKE sobre o conteúdo do campo Name, ou seja aquilo que o utilizador informa no campo Search name a aplicação interpretará como `SELECT ... FROM ... WHERE Name LIKE "%valor_campo_search_name%"`.

E ainda que é possível fazer a busca dos dados com combinações de conteúdos, utilizando os símbolos vírgula “,” (que representa o AND lógico) e ponto e vírgula “;” (que representa o OR lógico). Vejam os exemplos:

Para exibir qualquer conteúdo do primeiro semestre. Preencher Name com -1. A aplicação retornará 2006-1, 2007-1, 2008-1, 2009-1. A aplicação considerará o critério de busca LIKE `"%-1%"`.

1. Para exibir apenas 2008-1 e 2008-2. Preencher Name com 2008. A aplicação poderia retornar 2008-1, 2008-2, 2008-3, 2008-4, mas visto que só existirá 1 e 2. A aplicação considerará o critério de busca LIKE `"%2008%"`.
2. Para exibir apenas 2008-1 e 2009-1. Preencher Name com `2008-1;2009-1` ou a aplicação considerará o critério de busca LIKE `"%2008-1%"` OR `"%2009-1%"`.
3. Ao informar `2008-1,2009-1`, na esperança de obter o mesmo resultado do exemplo anterior, na verdade a aplicação não vai retornar resultados pois a expressão informada é equivalente a LIKE `"%2008-1%"` AND `"%2009-1%"`, e sabe-se que não existe um período com esta denominação.
4. Para voltar a exibir todos os dados de um módulo, basta fazer click no botão Search sem informar qualquer conteúdo na caixa Search Name.

Denotar que os critérios de pesquisa utilizados em um módulo se estendem para outro módulo, desde que este último utilize o primeiro. Por exemplo quando o critério de busca é definido no módulo Period, este critério se estende no módulo de Currentclass, uma vez que este último faz utilização do primeiro. Assim qualquer dos exemplos que foram mostrados acima para o módulo Period aparecerão no módulo Currentclass, quando for seleccionar em currentclass o period desejado.

8.5.5 Como adicionar um novo período

1. Click no link Period do menu. A aplicação o direccionará para módulo de períodos.
2. Click no botão Add. A aplicação o direccionará para a janela Add Period.
3. Preencha os conteúdos dos campos Name e Comments (facultativo).
4. Click no botão Submit. A aplicação fará a validação dos dados e os guardará (ou não conforme o resultado da validação).

Ilustração 20 – Adicionar novo período

Add Period

Name

Comments

8.5.6 Como alterar o conteúdo de um período

1. Estando no módulo `Period`, click no link `Edit` do período que deseja alterar os dados. A aplicação o direccionará para a janela `Edit Period`.
2. Altere os conteúdos dos campos `Name` e `Comments` (facultativo).
3. Click no botão `Submit`. A aplicação fará a validação dos dados e guardará as alterações (ou não conforme o resultado da validação).

Ilustração 21 – Alterar conteúdo de um período

Edit Period

Name

Comments

8.5.7 Como visualizar o conteúdo de um período

1. Estando no módulo `Period`, click no link `View` do período que deseja visualizar os dados. A aplicação o direccionará para a janela `Period`.
2. Click no botão `Back` se desejar voltar para a cena anterior ou no botão `Cancel`, se desejar ir para a página inicial do módulo de períodos.

Ilustração 22 – Visualizar conteúdo de um período

Period	
Id	5
Name	2008-1
Comments	Put here any comment
Created	2009-06-22 13:54:47
Modified	2009-07-10 12:21:04
Back	Edit Add Cancel

8.5.8 Como excluir períodos

1. Estando no módulo `Period`, click nos períodos que deseja excluir, marcando as respectivas `check boxes`.
2. Click no botão `Delete`. A aplicação fará a validação dos dados e os excluirá (ou não conforme o resultado da validação).

Ilustração 23 – Excluir períodos

Periods

Page 1 of 1, showing 9 records out of 9 total, starting on record 1

<input type="checkbox"/>	Id	Name	Comments
<input checked="" type="checkbox"/>	1 View Edit	2009-2	
<input type="checkbox"/>	2 View Edit	2009-1	
<input checked="" type="checkbox"/>	6 View Edit	2008-2	
<input checked="" type="checkbox"/>	5 View Edit	2008-1	Put here any comme
<input type="checkbox"/>	4 View Edit	2007-2	
<input checked="" type="checkbox"/>	3 View Edit	2007-1	
<input checked="" type="checkbox"/>	7 View Edit	2006-2	
<input type="checkbox"/>	8 View Edit	2006-1	teste
<input type="checkbox"/>	9 View Edit	2003-2	It's a comment

<< previous | next >>

Add Delete

8.5.9 Operações CRUD em outros módulos

Salvo as exceções apresentadas a seguir, para os demais módulos (ou seja Slots, Courses, Subjects, Typeclasses, Currentclasses, Group_Users, Users, Currentclass_Users, Rooms, Timetables e Slot_Users) as operações CRUD seguem o mesmo raciocínio do que foi apresentado para o módulo de períodos.

8.5.10 Como adicionar uma aula (currentclass)

Ilustração 24 – Adicionar uma aula

Add Currentclass

Period

2009-2
2009-1
2008-2
2008-1
2007-2

Course

DSIT
LDM
LEI
MCM
MEI

Subject

FM
GE
GPS
IAMB
IHC

Typeclass

PL1
PL2
PL3

Tot Slot

2

Comments

Submit Back Cancel Reset

1. Click no link `Currentclass` do menu. A aplicação o direccionará para módulo de aulas.
2. Click no botão `Add`. A aplicação o direccionará para a janela `Add Currentclass`.
3. Seleccione os `Period`, `Course`, `Subject` e `Typeclass`. Poderá fazer a selecção múltipla utilizando o click do rato mantendo carregada e a tecla `Ctrl` do teclado.
4. Preencha os conteúdos dos campos `Tot slot` e `Comments` (facultativo).
5. Click no botão `Submit`. A aplicação fará a validação dos dados e os guardará (ou não conforme o resultado da validação).

Para os módulos `Currentclass Users`, `Slot Users` e `Users` a forma de adicionar dados, segue raciocínio análogo.

8.5.11 Como adicionar um horário (timetable)

1. Click no link `Timetable` do menu. A aplicação o direccionará para módulo de horários.
2. Click no link `Add`. A aplicação o direccionará para a janela `Add Timetable`.
3. Seleccione a `Currentclass` desejada e click no botão `Next`.
4. Click mais uma vez no botão `Next` para a aplicação gerar as salas e slots disponíveis. Será exibida a grelha de slots para a escolha de quais slots e salas (`Room`) serão ocupadas para esta aula (`Currentclass`).
5. Seleccione os slots (click na respectiva `check box`) e sala (escolha uma das opções na `select box`). Convém verificar se tem slots disponíveis, para isto verifique se o conteúdo de “Slots for this `Currentclass`: Total 2, you have used 1, remained 1”. **Remained tem que ser maior que zero** para ser possível ocupar mais algum slot com a aula escolhida.
6. Preencha o conteúdo do campo `Comments` (facultativo).
7. Click no botão `Submit`. A aplicação fará a validação dos dados e os guardará (ou não conforme o resultado da validação).

Ilustração 25 – Adicionar um horário

Add Timetable

Currentclass
2008-2 MEI EPMult T1

Slots for this Currentclass: Total 2, you have used 1, remained 1.

	MON	TUE	WED
08:30-09:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
09:00-10:00	<input type="checkbox"/>	MEI EPMult (T1) B2	<input type="checkbox"/>
10:00-11:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11:00-12:00	<input type="checkbox"/>	MEI EPMult (PL1) C5.1	<input type="checkbox"/>
12:00-13:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13:00-14:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14:00-15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15:00-16:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16:00-17:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17:00-18:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18:00-19:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19:00-20:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comments

Submit Back Cancel Reset

[Add](#) [Delete](#) [List per Course](#) [List per Subject](#) [List per Room](#) [List per User](#)

8.5.12 Como excluir um horário (timetable)

1. Click no link `Timetable` do menu. A aplicação o direccionará para módulo de horários.
2. Click no link `Delete`. A aplicação o direccionará para a janela `Delete Timetable`.
3. Seleccione a `Currentclass` desejada e click no botão `Next`.
4. Seleccione os slots (click na respectiva `checkbox`).
5. Click no botão `Delete`. A aplicação fará a validação dos dados e os excluirá (ou não conforme o resultado da validação).

Ilustração 26 – Excluir um horário

Delete Timetable

Currentclass
2008-2 MEI EPMult T1

Slots for this Currentclass: Total 2, you have used 1, remained 1.

	MON	TUE
08:30-09:00		
09:00-10:00		<input type="checkbox"/> MEI EPMult (T1) B2
10:00-11:00		
11:00-12:00		<input type="checkbox"/> MEI EPMult (PL1) C5.1
12:00-13:00		
13:00-14:00		
14:00-15:00		
15:00-16:00		
16:00-17:00		
17:00-18:00		
18:00-19:00		
19:00-20:00		

Delete Back Cancel Reset

[Add](#) [Delete](#) [List per Course](#) [List per Subject](#) [List per Room](#) [List per User](#)