Mestrado em Engenharia Informática
Estágio
Relatório Final

CYCLOPS
Sistema Empresarial de Pesquisa

Carlos Manuel Fernandes dos Santos
carlosms@student.dei.uc.pt

Orientador DEI:
Marco Vieira

Orientador CRITICAL Software, SA:
Helder Sousa

Ano letivo: 2015/2016
Data: 15 de setembro de 2016

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

[Esta página foi intencionalmente deixada em branco]

Mestrado em Engenharia Informática
Estágio
Relatório Final

CYCLOPS
Sistema Empresarial de Pesquisa

Carlos Manuel Fernandes dos Santos
carlosms@student.dei.uc.pt

Orientador DEI:
Marco Vieira

Orientador CRITICAL Software, SA:
Helder Sousa

Júri:

Arguente:
Alexandre Miguel Pinto

Vogal:
Mario Rela

Ano letivo: 2015/2016
Data: 15 de setembro de 2016

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

[Esta página foi intencionalmente deixada em branco]

Resumo

Este relatório diz respeito ao desenvolvimento e implementação de um protótipo de um sistema empresarial de pesquisa, designado CYCLOPS, realizado no âmbito do estágio curricular que teve lugar na CRITICAL Software, SA empresa de desenvolvimento de software especializada em soluções, serviços e tecnologias para sistemas de informação críticos. O trabalho realizado, no âmbito deste estágio está, assim, relacionado com quatro áreas fundamentais associadas à pesquisa de informação, sendo elas: *search engine*; *information retrieval*; *enterprise search* e *bigdata*.

A revisão da literatura permitiu clarificar o problema associado ao desenvolvimento e implementação do CYCLOPS, transversal às plataformas de apoio à empresa, com foco no acesso/permissões de quem procura a informação e na relevância dos dados encontrados, não esquecendo os tempos de resposta e devendo a manutenção, necessária para manter este sistema operacional, ser a menor possível.

A abordagem ao desenvolvimento e implementação deste protótipo consistiu essencialmente na realização de um estudo exploratório, complementado por uma abordagem etnográfica inerente às características subjacentes a um estágio curricular. O estagiário foi integrado numa equipa de I&D podendo observar e contactar diretamente com a realidade empresarial em que foi desenvolvido o protótipo.

Durante o período de estágio para além de se ter desenvolvido e implementado o protótipo CYCLOPS procedeu-se à sua validação com recurso a testes manuais e automatizados: estes testes mostraram que o protótipo desenvolvido devolve, conforme esperado, resultados a partir de dois repositórios: o repositório *Subversion* (*SVN*) e o repositório *AsknowledgeDb*. Considerando que o desenvolvimento e implementação deste protótipo teve em atenção os desenvolvimentos mais recentes, na área da pesquisa de informação, o mesmo poderá ser melhorado com novas funcionalidades dando particular destaque ao paradigma *bigdata*.

O protótipo desenvolvido tem por objetivo facilitar a pesquisa de informação em repositórios corporativos, tendo em atenção a grande quantidade de informação existente nos referidos repositórios e a consequente dificuldade em encontrar informação pertinente e demonstra a viabilidade de desenvolver e implementar um sistema empresarial de pesquisa a que sejam acrescentadas outras funcionalidades.

Palavras-Chave

"*enterprise search*", "*search engine*", "*information retrieval*", "*bigdata*", "*solr*", "*lucene*", "*kafka*", "*crawler*", "*parser*", "*building blocks*", "*subversion*", "*mime types*"

[Esta página foi intencionalmente deixada em branco]

Abstract

This report details the development and implementation of a prototype of an enterprise search system, called CYCLOPS, conducted within the internship that took place in CRITICAL Software, SA a software development company specializing in solutions, services and technologies for critical information systems. The work carried out is thus related to four key areas associated with information Search: search engine; information retrieval; enterprise search and bigdata.

The state of the art has made it possible to clarify the problem associated with the development and implementation of this prototype, transverse to the platforms in support of the company, focusing on access/permissions for those who are looking for information and on the relevance of the data found, not forgetting the response times and maintenance, necessary to maintain this system, should be the smallest possible.

The approach to the development and implementation of this prototype was mainly focused in conducting an exploratory study, complemented by an ethnographic approach inherent of the characteristics underlying an internship. The trainee was integrated in a team of R&D allowing direct contact with the business reality in which the prototype was developed.

During the internship in addition to the development and implementation of a prototype, it was validated using manual and automated tests: these tests showed that the prototype developed returns, as expected, results from two repositories: the repository Subversion (SVN) and the repository AsknowledgeDb. Whereas the development and implementation of this prototype took into consideration the most recent developments in the area of information research, the same can be improved with new features giving particular emphasis to the bigdata paradigm.

The developed prototype is intended to facilitate the search for information in corporate repositories, taking into consideration the large amount of information available in these repositories and the consequent difficulty in finding relevant information and demonstrates the feasibility of developing and implementing an enterprise system of research to which are added other features.

Keywords

"*enterprise search*", "*search engine*", "*information retrieval*", "*bigdata*", "*solr*", "*lucene*", "*kafka*", "*crawler*", "*parser*", "*building blocks*", "*subversion*", "*mime types*"

[Esta página foi intencionalmente deixada em branco]

Índice

Capítulo 1 Introdução **1**
1.1 Contexto **1**
1.2 Problema **2**
1.2.1 Objetivos 2
1.2.2 Competências adquiridas 3
1.2.3 Abordagem 3
1.3 Estrutura do relatório **4**
Capítulo 2 Estado da Arte **5**
2.1 Enquadramento teórico **5**
2.1.1 Enterprise search 5
2.1.2 Information retrieval 7
2.1.3 Search engine 8
2.1.4 Bigdata 9
2.1.5 Tecnologia 11
2.2 Trabalho relacionado **12**
2.2.1 Recuperação de informação 12
2.2.2 Sistemas empresariais de pesquisa 13
2.2.3 Funcionalidades de pesquisa empresarial avançada 14
2.2.4 Plataformas open source 16
2.3 Tecnologias utilizadas **17**
2.3.1 Linguagem de programação Java 17
2.3.2 HyperText Markup Language - HTML 18
2.3.3 Cascading Style Sheets - CSS 18
2.3.4 Linguagem de programação JavaScript 18
2.3.5 Apache Tika 18
2.3.6 Apache Kafka 18
2.3.7 Apache Solr 19
2.3.8 Apache ZooKeeper 19
2.3.9 Apache Maven 19
2.3.10 Framework JUnit 19
Capítulo 3 Objetivos e Metodologia **21**
3.1 Clarificação do problema **21**
3.2 Abordagem metodológica **23**
3.2.1 Fase conceptual 23
3.2.2 Fase metodológica 24
3.3 Metodologia de desenvolvimento **26**
3.4 Ciclo de vida **27**
3.5 Planeamento **27**
3.5.1 Primeira fase 28
3.5.2 Segunda fase 28
3.6 Planeamento versus execução **30**

3.7 Gestão do risco **31**
3.7.1 Matriz de risco 32
3.7.2 Identificação e avaliação do risco 33
Capítulo 4 Requisitos e Arquitetura **39**
4.1 Especificação de requisitos **39**
4.1.1 Resultado das respostas ao questionário 40
4.1.2 Requisitos não funcionais 44
4.1.3 Requisitos funcionais 45
4.2 Elementos arquiteturais **52**
4.3 Building blocks **54**
4.3.1 Building blocks gerais 54
4.3.2 Building blocks diferenciadores 56
4.4 Arquitetura do protótipo **57**
Capítulo 5 Desenvolvimento do Protótipo **59**
5.1 Servidor de pesquisa empresarial **59**
5.2 Mecanismo de procura - crawler **60**
5.3 Mecanismo de transformação - parser **61**
5.4 Mecanismo de gestão de mensagens – Apache Kafka **62**
5.5 Mecanismo de indexação - indexer **63**
5.6 Lista de controlo de acesso **64**
5.7 Interface *Web* **64**
Capítulo 6 Testes e Verificação **67**
6.1 Funcionalidades testadas **67**
6.2 Funcionalidades não testadas **68**
6.3 Abordagem **68**
6.4 Crawler - casos de teste **69**
6.4.1 Quantidade de documentos armazenados no repositório 69
6.4.2 Tamanho dos documentos armazenados no repositório 69
6.4.3 Integridade dos documentos armazenados no repositório 70
6.4.4 Controlo de acesso aos documentos armazenados no repositório 70
6.5 Parser - casos de teste **71**
6.5.1 Extração de conteúdo 71
6.5.2 Documentos recebidos pelo parser 71
6.6 Indexer - casos de teste **72**
6.6.1 Documentos recebidos pelo indexer 72
6.6.2 Documentos indexados 72
6.7 Interface Web - casos de teste **73**
6.7.1 Caixa de pesquisa 73
6.7.2 Menu sidebar 73
6.7.3 Paginação 74
6.7.4 Logout 74
6.7.5 Login 74
6.7.6 Permissão/autorização 74
6.8 Teste de carga à interface Web **75**
Capítulo 7 Conclusões e propostas de trabalho futuro **79**
Referências **81**
ANEXOS **85**
Anexo 1 Lista de fornecedores enterprise search **87**
Anexo 2 Modelo de questionário **89**

Lista de figuras

Figura 1 - Principais componentes de um sistema de pesquisa de informação (Search Technologies, 2015b)..... 6
Figura 2 - Fluxograma representativo do processo *Information Retrieval* (Hearst, 1999)..... 7
Figura 3 - Estrutura dos *building blocks* associados a um motor de pesquisa (Croft et al., 2015)..... 9
Figura 4 - Desenvolvimento de um sistema de pesquisa de informação (Croft et al., 2015)..... 24
Figura 5 - Áreas de suporte, grandes consumidores de informação empresarial (Leher, 2014)..... 25
Figura 6 – Plasticidade associada às metodologias ágeis/tradicionais (Barbosa et al., 2007)..... 26
Figura 7 - Ciclo de vida da desenvolvimento incremental 27
Figura 8 - Diagrama de Gantt relativo às atividades realizadas durante o primeiro semestre..... 28
Figura 9 - Diagrama de Gantt representativo das atividades a desenvolver durante o segundo semestre..... 29
Figura 10 - Diagrama de *Gantt* relativo à execução do trabalho desenvolvido na segunda fase do projeto 30
Figura 11 - Componentes do risco (Alberts & Dorofee, 2010) 31
Figura 12 - Etapas do processo de gestão do risco (Alberts & Dorofee, 2010)..... 31
Figura 13 - Arquitetura geral de um sistema de pesquisa de informação (Koren, 2013) 52
Figura 14 - Arquitetura de um sistema de pesquisa de informação com *bigdata* (Search Technologies, 2015a).. 53
Figura 15 - Processo de indexação de um motor de busca (Croft et al., 2015) 53
Figura 16 - Processo de procura de um motor de busca (Croft et al., 2015) 54
Figura 17 - Arquitetura geral de um sistema de pesquisa de informação empresarial..... 54
Figura 18 - Arquitetura do protótipo do sistema CYCLOPS..... 57
Figura 19 - Componentes de um sistema empresarial de pesquisa com integração da plataforma Apache Solr 60
Figura 20 - Bloco responsável pela procura e recolha de dados dos repositórios *SVN* e *AcknowledgeDb*..... 60
Figura 21 - Apache Kafka como plataforma gestora de mensagens entre produtor e consumidor..... 61
Figura 22 - Bloco responsável pela transformação dos documentos recolhidos no repositório *SVN*..... 61
Figura 23 - Bloco responsável pelo sistema de gestão de mensagens 62
Figura 24 - Mecanismo de troca de mensagens organizadas por tópicos..... 63
Figura 25 - Bloco responsável pelo processo de indexação..... 63
Figura 26 - Bloco responsável pela interface Web 64
Figura 27 - Interface Web do protótipo do sistema CYCLOPS..... 65
Figura 28 - Resultados do teste de carga para 25 utilizadores 75
Figura 29 - Resultados do teste de carga para 50 utilizadores 75
Figura 30 - Resultados do teste de carga para 75 utilizadores 76
Figura 31 - Resultados do teste de carga para 100 utilizadores 76

[Esta página foi intencionalmente deixada em branco]

Lista de gráficos

Gráfico 1 - Evolução do número de sites, no início da *Web* 8
Gráfico 2 - Evolução previsível do volume de informação digital até 2020 (Gantz & Reinsel, 2010) 9
Gráfico 3 - Confirmação da evolução do volume de informação digital até 2020 (Turner et al., 2014) 10
Gráfico 4 - Língua utilizada com mais frequência na pesquisa de informação empresarial 40
Gráfico 5 - Elemento considerado mais relevante na pesquisa de informação empresarial..... 41
Gráfico 6 - Tipo de filtro considerado mais relevante na pesquisa de informação empresarial 41
Gráfico 7 - Critérios que podem ser uma mais valia no âmbito da pesquisa de informação empresarial 42
Gráfico 8 - Tipo de informação usada como critério de pesquisa de informação empresarial..... 42
Gráfico 9 - Tipo de documentos mais procurados no âmbito da pesquisa de informação empresarial..... 43
Gráfico 10 - Importância de um mecanismo de apoio à pesquisa de informação empresarial..... 43
Gráfico 11 - Relevância atribuída à antiguidade dos documentos a procurar..... 44

[Esta página foi intencionalmente deixada em branco]

Lista de quadros

Quadro 1 - Código que implementa o caso de teste "quantidade de documentos" 69
Quadro 2 - Código que implementa o caso de teste "tamanho dos documentos" 69
Quadro 3 - Código que implementa o caso de teste "integridade dos documentos" 70
Quadro 4 - Código que implementa o caso de teste "controlo de acesso aos documentos" 70
Quadro 5 - Código que implementa o caso de teste "extração de conteúdo dos documentos" 71
Quadro 6 - Código que implementa o caso de teste "documentos recebidos pelo *parser*" 71
Quadro 7 - Código que implementa o caso de teste "documentos recebidos pelo *indexer*" 72
Quadro 8 - Código que implementa o caso de teste "documentos indexados" 72

[Esta página foi intencionalmente deixada em branco]

Lista de tabelas

Tabela 1 - Escala relativa á probabilidade de ocorrência (Bonanomi et al., 2012) 32
Tabela 2 - Escala relativa ao impacto que pode existir (Bonanomi et al., 2012) 32
Tabela 3 - Matriz de risco (probabilidade x impacto) (Bonanomi et al., 2012) 33
Tabela 4 - Identificação e priorização de riscos que podem ocorrer no decurso do *Sprint1* 33
Tabela 5 - Identificação e priorização do risco que pode ocorrer no período relativo ao *Sprint2* 34
Tabela 6 - Identificação e priorização de riscos que podem ocorrer no período relativo ao *Sprint3* 35
Tabela 7 - Identificação e priorização de riscos que podem ocorrer no período relativo ao *Sprint4* 35
Tabela 8 - Identificação e priorização de riscos que podem ocorrer no decurso do *Sprint2* 36
Tabela 9 - Requisito funcional (SR - 1.01.00 - identificação do repositório) 46
Tabela 10 - Requisito funcional (SR - 1.01.01 - repositório *file based*) 46
Tabela 11 - Requisito funcional (SR - 1.01.02 - repositório *database record based*) 46
Tabela 12 - Requisito funcional (SR - 1.02.00 - mecanismo de procura) 47
Tabela 13 - Requisito funcional (SR - 1.02.01 - alteração do estado do repositório) 47
Tabela 14 - Requisito funcional (SR - 1.02.01.01 - adição de novos documentos ao repositório) 47
Tabela 15 - Requisito funcional (SR - 1.02.01.02 - exclusão de documentos do repositório) 47
Tabela 16 - Requisito funcional (SR - 1.02.01.03 - atualização de documentos no repositório) 48
Tabela 17 - Requisito funcional (SR - 1.02.02 - procurar documentos por *MIME type*) 48
Tabela 18 - Requisito funcional (SR - 1.02.03 - associar ACL à procura de documentos) 48
Tabela 19 - Requisito funcional (SR - 1.02.04 - gestão de documentos) 48
Tabela 20 - Requisito funcional (SR - 1.02.05 - output do mecanismo de procura) 48
Tabela 21 - Requisito funcional (SR - 1.03.00 - fila de documentos para transformar) 49
Tabela 22 - Requisito funcional (SR - 1.04.00 - fila de documentos para indexar) 49
Tabela 23 - Requisito funcional (SR - 1.05.00 - mecanismo de transformação) 49
Tabela 24 - Requisito funcional (SR - 1.05.01 - extração de informação do conteúdo de um documento) 49
Tabela 25 - Requisito funcional (SR - 1.05.02 - extração de metadados) 50
Tabela 26 - Requisito funcional (SR - 1.06.00 - mecanismo de indexação) 50
Tabela 27 - Requisito funcional (SR - 2.01.00 - pesquisa de documentos) 50
Tabela 28 - Requisito funcional (SR - 2.02.00 - filtrar documentos) 51
Tabela 29 - Requisito funcional (SR - 2.03.00 - paginação) 51
Tabela 30 - Requisito funcional (SR - 2.04.00 - relevância) 51
Tabela 31 - Requisito funcional (SR - 2.05.00 - *logout*) 51
Tabela 32 - Requisito funcional (SR - 2.06.00 - *login*) 51
Tabela 33 - Resultados do caso de teste "quantidade de documentos" 69
Tabela 34 - Resultados do caso de teste "tamanho dos documentos" 70
Tabela 35 - Resultados do caso de teste "integridade dos documentos" 70
Tabela 36 - Resultados do caso de teste "controlo de acesso aos documentos" 71
Tabela 37 - Resultados do caso de teste "extração de conteúdo dos documentos" 71
Tabela 38 - Resultados do caso de teste "documentos recebidos pelo *parser*" 71
Tabela 39 - Resultados do caso de teste "documentos recebidos pelo *indexer*" 72
Tabela 40 - Resultados do caso de teste "documentos indexados" 72

[Esta página foi intencionalmente deixada em branco]

Capítulo 1
Introdução

O presente relatório é relativo ao estágio curricular realizado, no decorrer do ano letivo 2015/2016, no âmbito do desenvolvimento e implementação do protótipo de um sistema empresarial de pesquisa, designado CYCLOPS, que teve lugar na empresa CRITICAL Software, SA. Este estágio teve como orientadores o Professor Doutor Marco Vieira do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra e o Engenheiro Helder Sousa colaborador da CRITICAL Software, SA uma empresa de desenvolvimento de software especializada em soluções, serviços e tecnologias para sistemas de informação críticos associados a organizações com atuação nos mais diversos setores de atividade.

1.1 Contexto

A enorme quantidade de informação que é atualmente produzida em qualquer tipo de organização, pública ou privada, tem impacto significativo na dificuldade que se coloca a qualquer dos seus colaboradores, que pretenda encontrar determinada informação. Esta procura de informação é responsável por grande desperdício de tempo nas organizações (PengFei (Vincent), Thomas, & Hawking, 2013), com efeito relevante na sua produtividade e consequentemente nos seus índices de desempenho.

Um sistema empresarial de pesquisa (*enterprise search*) descreve, genericamente, um conjunto de aplicações utilizadas para procurar e recuperar informação empresarial, com foco em repositórios corporativos, usando diferentes tecnologias. Estes sistemas podem ser preciosos auxiliares dos colaboradores de uma determinada organização ajudando-os, de diversas formas, designadamente: a encontrar a informação necessária e relevante para a sua atividade profissional e a apresentar os resultados que melhor satisfaçam as suas necessidades laborais podendo, assim, dar um forte contributo para minimização do desperdício de tempo, na procura de informação empresarial, e consequentemente dos custos associados aos diversos processos produtivos.

No entanto, por diversas razões, a tarefa associada à pesquisa de informação empresarial pode tornar-se bastante complexa, devido à especificidade associada a determinados repositórios corporativos, constituindo-se esta situação num problema que carece de resposta por parte das organizações. Por esta razão, estas, procuram incessantemente encontrar as respostas mais adequadas que ajudem a melhorar os sistemas empresariais de pesquisa.

1.2 Problema

Presentemente, na linha do que foi referido anteriormente, na CRITICAL Software, SA não existe um sistema empresarial de pesquisa, suficientemente habilitado para efetuar pesquisas, considerando toda a informação armazenada nos seus repositórios corporativos, nomeadamente: o repositório *subversion SVN* e o repositório *AsknowledgeDb*. Este facto pode originar diversos constrangimentos uma vez que dificulta a procura de informação ou simplesmente saber se existe ou não nos repositórios da empresa. Esta situação pode levar à duplicação de informação, ou à falta de apoio à tomada de decisão mais adequada, devido ao tempo que pode ser necessário despender para obter a informação necessária.

Face ao exposto, anteriormente, este estágio tem como objetivo o desenvolvimento de um protótipo do sistema CYCLOPS, capaz de encontrar um conjunto limitado de *MIME types* e proceder à sua indexação. Paralelamente às operações de pesquisa e indexação o protótipo disponibilizará uma interface *Web* para que os utilizadores possam interagir obtendo os resultados considerados mais relevantes, tendo em conta as credenciais de cada utilizador.

1.2.1 Objetivos

Assim, foi desenvolvido e implementado um protótipo do sistema CYCLOPS numa abordagem *in-house*, com o objetivo de recolher e indexar informação armazenada, especificamente, em dois repositórios corporativos:

- o repositório *subversion* (*SVN*) associado ao sistema de controlo de versões. Este repositório é relevante para o processo produtivo da CRITICAL Software, SA; e,
- o repositório *AsknowledgeDb* associado a uma base de dados que contém os dados de todos os colaboradores da CRITICAL Software, SA. Conforme se pode entender este repositório é igualmente relevante para a organização.

A opção por estes dois repositórios ficou a dever-se, obviamente, à sua importância e também à limitação de tempo existente que condicionou, nesta fase, a inclusão de outros repositórios da empresa. O desenvolvimento deste protótipo baseou-se no recurso a diversas tecnologias que deem garantias de indexação e pesquisa rápida e relevante.

Embora exista um grande número de ficheiros em cada um dos repositórios, referidos anteriormente, com diversos formatos (*MIME type* – identificador padrão usado na Internet para indicar o tipo de dados que um determinado ficheiro contém[1]), também por razões de tempo foi dada particular atenção à indexação de documentos do tipo word, powerpoint, pdf, plain text e registos de base de dados.

Entre os diversos desafios que a realização deste protótipo colocou, merecem destaque aqueles que foram a base dos dois grandes objetivos da realização deste estágio curricular:

- o primeiro objetivo relacionado com a necessidade de desenvolver competências para recolher, indexar e pesquisar dados em repositórios corporativos heterogéneos, o que implica a não existência de interfaces comuns entre os repositórios dificultando a procura nos mesmos; e,
- o outro grande objetivo relacionado com a gestão das permissões de acesso de cada utilizador, por documento, baseada numa lista de controlo de acessos (*Access Control List* (*ACL*)). A concretização deste objetivo é bastante pertinente e relevante visto que não há no mercado oferta de soluções que, de uma maneira simples e eficiente, façam a gestão da lista de controlo de acessos em diferentes repositórios.

[1] Originalmente estes identificadores foram caracterizados pela RFC 2046, tendo posteriormente a sua utilização sido expandida através da RFC 6838. *Request for Comments* (RFC)

1.2.2 Competências adquiridas

Conforme referido anteriormente, a realização do presente estágio curricular apresentou vários desafios. A resolução destes desafios proporcionou ao estagiário a aquisição de uma visão real do mundo empresarial, por ter integrado uma equipa a operar no terreno, e também o desenvolvimento de *know-how* nalgumas áreas relevantes no âmbito da pesquisa de informação, nomeadamente, as seguintes:

- tecnologias disponibilizadas pela Apache Software Foundation e do tipo J2EE, para além de técnicas de recolha, indexação e pesquisa de informação emprearial;
- desenho de arquiteturas adequadas às tecnologias referidas anteriormente, assim como o seu desenvolvimento e implementação;
- Identificação e especificação de requisitos, bem como a realização de testes e verificação de sistemas de software; e,
- processos de engenharia de software (p. ex. gestão, *quality assurance* e desenvolvimento de software).

Para realizar este estágio curricular o estagiário foi integrado na equipa I&D da área *Data & Business Analytics* da empresa iTGROW[2] vocacionada para formação e treino de competências *on-the-job*. O estágio teve como objetivo principal, conforme já referido anteriormente, o estudo, análise, especificação, desenvolvimento, implementação e validação de um protótipo para um sistema de pesquisa de informação empresarial. Este sistema tem a designação CYCLOPS e deverá ser utilizado, internamente, pelos colaboradores da CRITICAL Software, SA para pesquisarem informação necessária à sua atividade profissional.

1.2.3 Abordagem

A abordagem utilizada para dar resposta aos desafios colocados durante o processo de estudo, desenvolvimento e implementação do presente protótipo. No essencial, este processo foi divido em duas fases: uma primeira fase que decorreu entre setembro de 2015 e janeiro de 2016, primeiro semestre do ano letivo 2015/2016, e uma segunda fase que decorreu entre fevereiro e junho de 2016, segundo semestre do ano letivo 2015/2016.

A primeira fase do estágio foi dedicada à recolha de informação sobre o que, de facto, se pretendia. Para isto procedeu-se a uma revisão bibliográfica, não só ao nível dos conceitos associados às principais áreas associadas à recolha, indexação e pesquisa de informação empresarial, armazenada em repositórios corporativos, a saber: *enterprise search*; *information retrieval*; *search engine*; *bigdata* e obviamente as tecnologias associadas, mas, também na procura e revisão de trabalho relacionado com a pesquisa de informação empresarial, tendo sido analisados vários estudos realizados entre 2008 e 2015 com particular incidência nos últimos 5 anos. Foi, igualmente, nesta fase que se iniciou o processo de levantamento e especificação de requisitos não funcionais e funcionais e também de atributos de qualidade a que deverá obedecer o desenvolvimento de um protótipo do sistema CYCLOPS.

A segunda fase do estágio foi dedicada ao desenvolvimento e implementação do protótipo. Nesta fase foram identificados os componentes necessários para proceder ao referido desenvolvimento e implementação e a opção a tomar em termos de tecnologia: aplicações comerciais ou tecnologias *open source*?. Alguns dos módulos do protótipo foram desenvolvidos de raiz, recorrendo à linguagem Java e outros foram implementados em plataformas *open source*.

[2] http://www.itgrow.pt/pt/inicio

1.3 Estrutura do relatório

O estágio curricular, referido anteriormente, implicou a realização de várias atividades que são descritas, na generalidade, nos capítulos que constituem o presente relatório que se encontra estruturado num total de 7 capítulos e 2 anexos.

No Capítulo 1 é feita a introdução que contextualiza e apresenta as razões que motivaram a realização deste protótipo, sendo também feita a descrição genérica do conteúdo do presente relatório.

No Capítulo 2 apresenta-se a revisão bibliográfica que foi necessário fazer para que se percebessem os principais conceitos associados à pesquisa de informação empresarial, assim como a identificação e análise de trabalhos relacionados que permitiram tomar noção do ponto em que se encontra o estado da arte da pesquisa de informação empresarial, focada na pesquisa, recolha e indexação de documentos armazenados em repositórios corporativos.

No Capítulo 3 é apresentado, de forma mais detalhada o problema que foi identificado, no âmbito do projeto associado ao estágio curricular realizado na CRITICAL Software, SA e a que deve ser dada adequada resposta assim como a identificação dos objetivos que se pretendem atingir, com a realização deste projeto. É igualmente apresentada a abordagem metodológica utilizada e as razões que levaram às opções tomadas. Finalmente, ainda neste capítulo, é apresentado o planeamento que orientou a realização do presente estágio curricular e é feita para cada um dos *Sprints* a respetiva identificação e avaliação do risco, com indicação dos mecanismos de controlo que devem ser implementados para minimização dos referidos riscos, com base na análise modal de falhas.

O Capítulo 4 é destinado à especificação dos requisitos, não funcionais e funcionais, a que deve obedecer o desenvolvimento e implementação do protótipo do sistema CYCLOPS assim como ao desenho da arquitetura que se entenda ser a mais adequada para se obter uma boa solução que dê resposta adequada ao problema clarificado no capítulo 3.

No Capítulo 5 são apresentados de forma detalhada alguns dos procedimentos associados às várias fases do processo de desenvolvimento e implementação do protótipo realizado.

No Capítulo 6 são apresentados os vários testes e verificações feitos ao protótipo do sistema CYCLOPS. Os testes ao sistema propriamente dito são automatizados, enquanto que os testes feitos á interface *Web* são manuais.

Finalmente no Capítulo 7 são apresentadas as principais conclusões e propostas que poderão dar origem a outros desenvolvimentos, no âmbito de trabalhos futuros.

Em anexo é apresentado o Anexo 1 que contém uma lista de fornecedores de sistemas de pesquisa de informação e o Anexo 2 que apresenta o modelo do questionário utilizado para perceber o sentimento dos colaboradores da iTGROW e da CRITICAL Software, SA no âmbito do desenvolvimento e implementação de um sistema empresarial de pesquisa.

Capítulo 2
Estado da Arte

Para que o estagiário pudesse adquirir e desenvolver as competências necessárias foi necessário proceder a uma adequada revisão bibliográfica focada, essencialmente, nas áreas referidas anteriormente, podendo assim perceber o estado da arte e o trabalho relacionado existente e assim detalhar o problema que se coloca no âmbito do desenvolvimento e implementação de um protótipo do sistema CYCLOPS.

Nas secções seguintes percebe-se o enquadramento teórico associado a cada uma das áreas específicas, referidas anteriormente, é feito o ponto de situação em que se encontra o estado da arte e o trabalho relacionado no domínio dos sistemas empresariais de pesquisa e, finalmente, é feita uma rápida apreciação às tecnologias utilizadas no âmbito do desenvolvimento e implementação do protótipo do sistema CYCLOPS.

2.1 Enquadramento teórico

No âmbito do presente estágio curricular foi identificada a necessidade de serem adquiridas e desenvolvidas, por parte do estagiário, competências específicas nas áreas de: *search engine*; *information retrieval*; *enterprise search* e ainda na utilização de novas tecnologias no âmbito do paradigma *bigdata* para que se possam atingir os objetivos genéricos definidos no âmbito de um sistema empresarial de pesquisa de informação.

2.1.1 Enterprise search

O termo *enterprise search* é, tipicamente, usado para descrever genericamente aplicações que pesquisam e recuperam informação empresarial usando diferentes tecnologias com foco na procura em repositórios corporativos (White & Nikolov, 2013). White (2012, p. 5) define *enterprise search* como "*an enterprise search application enables employees to find all the information that the company possesses without the need to know where the information is stored*".

O mesmo autor, White (2012, p. 6), ainda na mesma obra, afirma que "*information retrieval deals with the representation, storage, organization of and access to information items such as documents, Web pages, online catalogues, structured and semi-structured records and multimedia objects*" e chama a atenção para o *gap* que existe, entre as competências oferecidas pelos cursos de graduação na área das tecnologias de informação e comunicação, e as competências necessárias à implementação de tecnologias relacionadas com *information retrieval*. Em contrapartida, deve referir-se que muitos dos principais fornecedores de tecnologias de informação e comunicação, tais como a IBM, a Oracle, a HP, a Google e a Microsoft, têm uma longa história de realizações conseguidas com sucesso no âmbito da pesquisa de informação empresarial.

Softic, Rosemberger, Zoier, Mondelos, e Pillinger (2013) levaram a cabo um estudo, como primeiro passo, no sentido de identificarem os principais motores de pesquisa, utilizados na

pesquisa de informação empresarial, como facilitadores na resolução dos desafios colocados relativamente ao *Workplace*, tais como: transferência de conhecimento e gestão individual e organizacional da informação no âmbito da colaboração corporativa. Este estudo concluiu que todos os motores de pesquisa avaliados mostraram uma elevada capacidade de integração e modularização.

Na Figura 1 são mostrados os principais componentes que fazem parte de um sistema de pesquisa de informação empresarial. Esta representação tem-se mantido bastante estável ao longo dos últimos anos, razão pela qual aqui é reproduzida (Search Technologies, 2015b). De acordo com White e Nikolov (2013) os sistemas de pesquisa empresarial remontam ao final dos anos 60 do século XX, quando houve necessidade de desenvolver sistemas para procurar informação científica, comercial e jurídica em grandes bases de dados para apoiar equipas jurídicas *anti-trust* nos Estados Unidos da América (EUA).

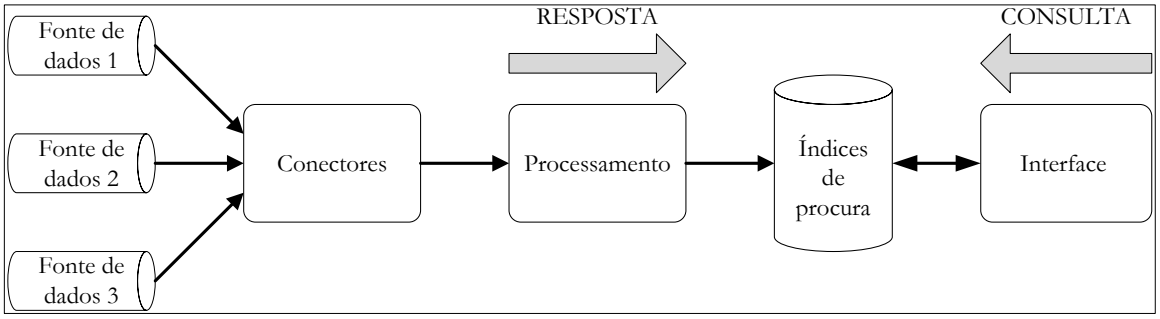


Figura 1 - Principais componentes de um sistema de pesquisa de informação (Search Technologies, 2015b)

Conforme referido, através de um documento da responsabilidade da Google (2006), os sistemas de pesquisa de informação empresarial não são diferentes daquilo que acontece em muitas outras categorias, no âmbito das tecnologias da informação e comunicação, em que os fornecedores têm por hábito atribuir diferentes nomes a recursos similares que podem ser oferecidos por diversos dos seus produtos. Esta situação como é fácil de entender pode ser geradora, muitas vezes, de uma boa dose de confusão junto de potenciais clientes.

Ainda no mesmo documento, (Google, 2006), é admitido que, em síntese, tudo se pode resumir à análise de sete fatores críticos quando é sentida necessidade por alguma organização de proceder à implementação de uma solução de pesquisa de informação empresarial. Os sete fatores referidos são os seguintes: (1) relevância; (2) experiência do utilizador; (3) âmbito; (4) atualidade da informação; (5) controlo de acessos; (6) escalabilidade e (7) custo total.

Hawking (2004) refere que um dos grandes problemas associados à pesquisa de informação empresarial, que urge resolver, está relacionado com os seus custos. A Oracle refere que os custos que as organizações suportam devido a má gestão dos seus dados são bastante elevados, estimando-se que as organizações possam perder anualmente, em média, 14% das suas receitas devido à ineficácia na gestão da sua informação[3].

Podemos, assim, concluir que o grande desafio da pesquisa de informação empresarial consiste em proporcionar às organizações um melhor conhecimento através de consultas, feitas adequadamente, aos repositórios corporativos onde reside informação que interessa às organizações. Assim, um sistema de pesquisa de informação empresarial deve ter capacidade para poder pesquisar e recolher documentos de diversos repositórios corporativos convertendo-os para formatos adequados (Hawking, 2010).

[3] Source: Oracle Survey, *From Overload to Impact: an industry scorecard on big data business challenges*, 2012

2.1.2 Information retrieval

O processo *information retrieval* pode ser descrito por uma sequência de passos representados no fluxograma mostrado na Figura 2. Este fluxograma foi apresentado por Hearst (1999).

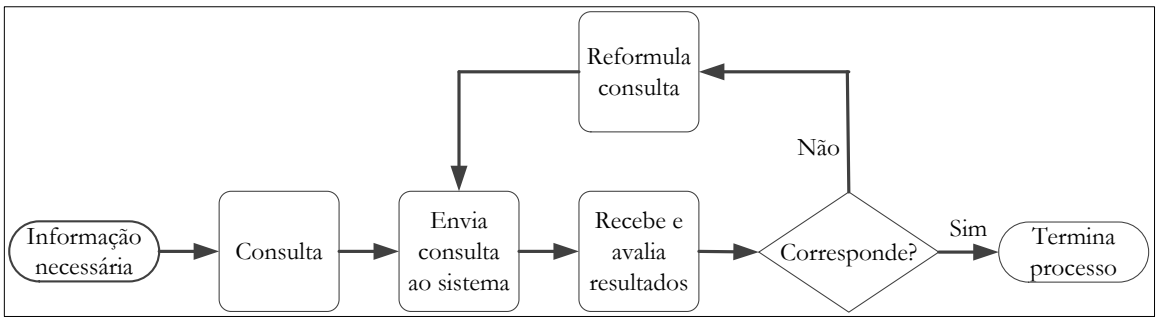


Figura 2 - Fluxograma representativo do processo *Information Retrieval* (Hearst, 1999)

Information retrieval não é propriamente uma área de investigação recente, Swanson (1961) já tecia diversas considerações sobre o estado da arte relativamente ao paradigma *information retrieval*. Esta temática está relacionada com o paradigma associado à pesquisa de informação empresarial representada em qualquer formato e disponibilizada em qualquer lugar (Lew, Sebe, Djeraba, & Jain, 2006).

O objetivo do paradigma *information retrieval* consiste em fornecer aos diversos interessados os documentos que, de alguma forma, satisfaçam a sua necessidade de informação empresarial. Esta necessidade de informação é normalmente expressa em linguagem natural, que nem sempre é bem estruturada e, muitas vezes, é semanticamente ambígua (Nunes, 2006). Assim, a implementação de sistemas de pesquisa de informação que tenham por objetivo a pesquisa de informação empresarial deve ter presente a forma pouco estruturada daquela informação.

O sucesso da *Web* nos últimos anos forçou uma significativa mudança de paradigma no âmbito da *information retrieval* devido, entre vários outros fatores, à democratização da publicação de conteúdos (Hawking, 2004). Assim, hoje, o paradigma *information retrieval* referindo-se em abstrato ao mesmo conceito, pesquisa de informação empresarial, deve ser abordado de forma diferente.

Em junho de 1993, conforme refere Gray[4], existiam cerca de 130 *websites*, em dezembro desse mesmo ano o número de *websites* havia crescido para 623 e seis meses depois, verificava-se novo acréscimo para 2738, mantendo-se esta tendência. Para fazer face a este crescimento exponencial, ver Gráfico 1, começaram a surgir os motores de busca da *Web*, no final de 1993, que vieram colocar novos desafios ao paradigma *information retrieval* (Sanderson & Croft, 2012).

[4] http://www.mit.edu/~mkgray/net/web-growth-summary.html

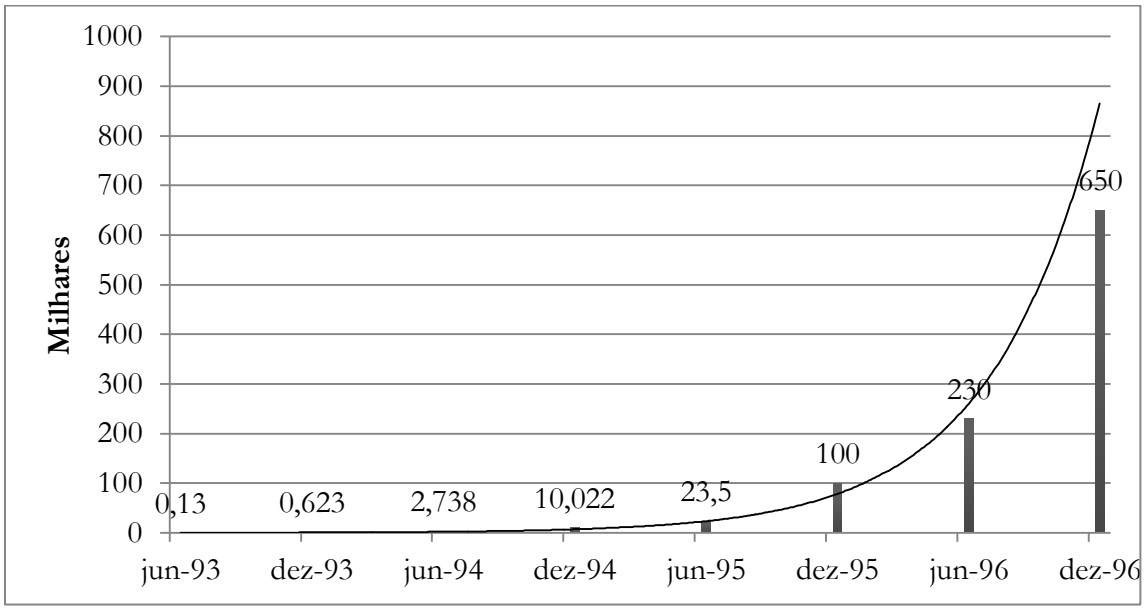


Gráfico 1 - Evolução do número de sites, no início da *Web*[5]

2.1.3 Search engine

De acordo com Croft, Donald Metzler e Strohman (2015, p. 6) "*a search engine is the practical application of information retrieval techniques to large-scale text collections*", podendo ser encontrado em grande variedade de aplicações (*desktop search* ou *enterprise search*).

Os motores de pesquisa de código aberto (*open source*) são uma classe importante que tem poucas diferenças relativamente aos motores de pesquisa comerciais (Croft et al., 2015), sendo portanto uma boa alternativa a estes. Podem ser destacados três importantes *search engine* de código aberto: Lucene[6], Lemur[7] e Galago[8].

Tipicamente, os componentes de um motor de pesquisa estão associados a dois grandes processos por ele excutados: (1) processo de indexação e (2) processo de consulta. O processo de indexação tem por função criar uma estrutura que permita executar e otimizar a procura de informação. Por sua vez o processo de consulta, utilizando a estrutura criada pelo processo de indexação juntamente com a consulta feita pelo utilizador, produz uma lista ordenada, de acordo com algum critério, da informação procurada (Croft et al., 2015).

Estes processos são constituídos por tarefas. Assim, o processo de indexação é composto por: aquisição do texto; transformação do texto e criação dos índices e o processo de consulta é composto por: interação com o utilizador; *ranking* e avaliação, ver Figura 3. A estrutura constituída por processos, tarefas e componentes é designada genericamente de *building blocks*, elementos muito importantes, que permitem que o desenvolvimento de sistemas de pesquisa de informação empresarial sejam parametrizados em função dos requisitos identificados junto de quem vai utilizar o referido sistema (Croft et al., 2015).

[5] http://www.mit.edu/~mkgray/net/web-growth-summary.html
[6] http://lucene.apache.org
[7] http://www.lemurproject.org
[8] http://www.search-engines-book.com

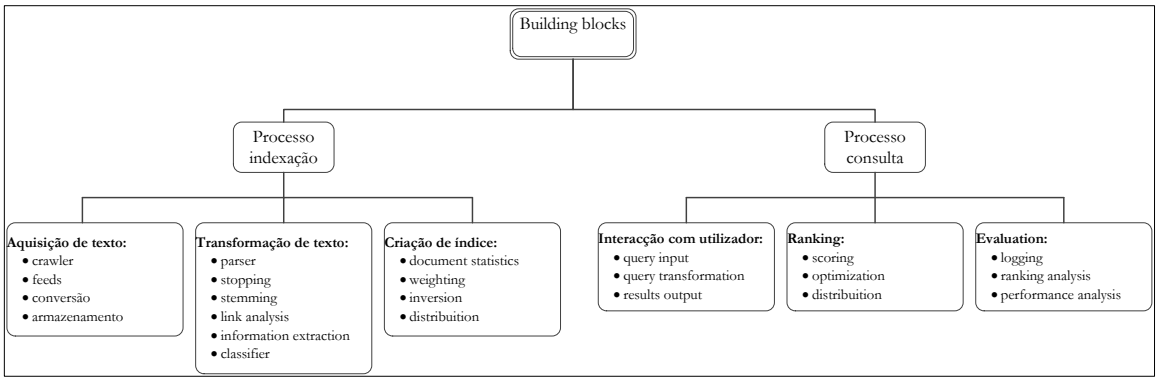


Figura 3 - Estrutura dos *building blocks* associados a um motor de pesquisa (Croft et al., 2015)

2.1.4 Bigdata

A importância do *bigdata* e a tendência que se está a verificar na mudança de paradigma nesta direção, pode ser vista todos os dias. Grantz e Reinsel (2010) dizem que até 2020 o volume de informação digital, criada e replicada por todo o mundo, vai crescer de forma exponencial até ao patamar dos inconcebíveis, há alguns anos atrás, 35 Zetabyte[9] incluindo as principais formas de média (voz, TV, rádio, imprensa) no formato digital, Gráfico 2.

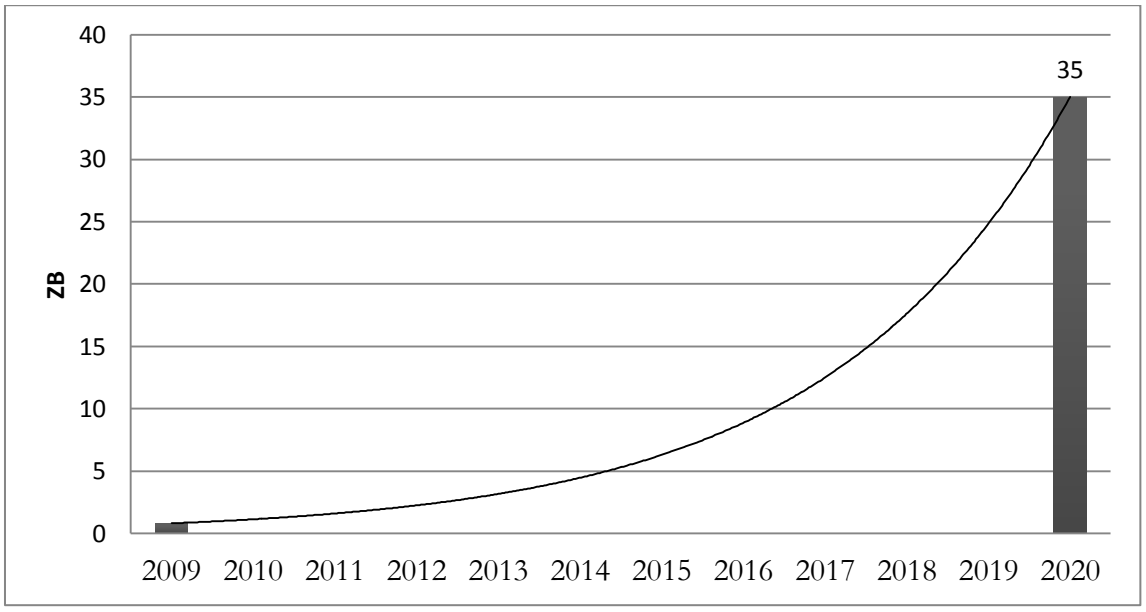


Gráfico 2 - Evolução previsível do volume de informação digital até 2020 (Gantz & Reinsel, 2010)

Esta tendência é confirmada, em alta, por Turner, Gantz, Reinsel e Minton (2014), conforme apresentado na Gráfico 3. Assim, segundo Chen e Zhang (2014) não há dúvida de que, futuramente, a produtividade e as tecnologias associadas aos negócios convergirão necessariamente para exploração de grandes volumes de dados (*bigdata*). Este facto, é razão para que qualquer estudo relacionado com pesquisa de informação empresarial, desenvolvido atualmente, deva ter em atenção esta realidade.

[9] Zetabyte = 1 trilião de Gigabyte

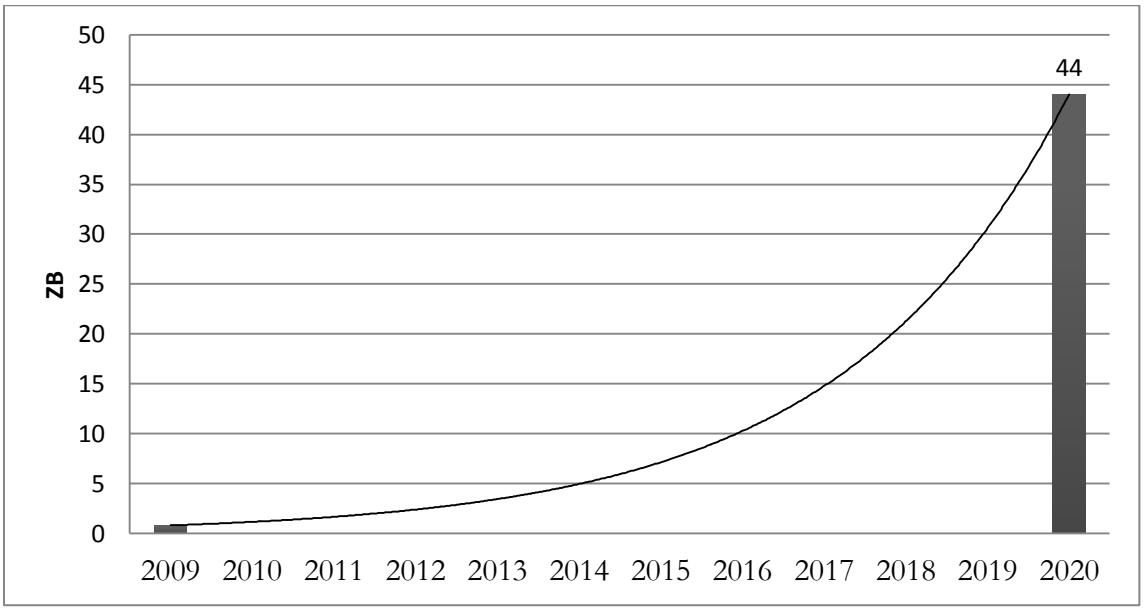


Gráfico 3 - Confirmação da evolução do volume de informação digital até 2020 (Turner et al., 2014)

Li, Liu e Zhu (2014) discutem os grandes desafios que, em sua opinião, se colocarão futuramente no desenvolvimento e implementação de sistemas de pesquisa empresarial, designadamente:

- é aconselhável uma análise dos dados em profundidade, para além da simples e tradicional pesquisa por palavras-chave, isto poderá alavancar uma melhor pesquisa de dados empresariais;
- uma outra questão que se pode colocar está relacionada com as tecnologias para pesquisa de *bigdata* de forma a serem incorporadas técnicas de análise sofisticadas que apoiem a pesquisa de informação empresarial de forma escalável;
- na opinião destes autores colocam-se muitos outros desafios, mas a grande maioria está relacionada com os *bigdata* e a forma escalável da pesquisa de informação empresarial; e,
- os autores, acreditam que apesar dos avanços recentes em *enterprise search* os motores de pesquisa empresariais ainda permanecem geridos de forma *adhoc*.

No relatório da Search Technologies (2015a) podemos ler que durante os próximos anos, muitas das melhorias substanciais a incorporar nos sistemas de pesquisa de informação empresarial incidirão na área do processamento de texto. Uma pesquisa de informação empresarial que faça a indexação de dez milhões de documentos estará a processar dezenas de biliões de palavras e frases, fazendo com que a análise do texto envolva uma grande quantidade de dados.

Tem-se verificado que vários vendedores de soluções *enterprise search* têm vindo a apostar seriamente na sua adequação ao paradigma *bigdata* (White, 2012), designadamente: Attivio; Autonomy; LucidWorks; MarkLogic; Sinequa; Coveo e Funnelback.

O relatório da Transparency Market Research (2014) inclui informação relevante sobre os principais *players* no mercado, designadamente a IBM Corporation, Microsoft Corporation, Oracle Corporation, SAP AG, Attivio Software Incorporation, Coveo Corporation, Dassault Systems SA, MarkLogic Corporation e X1 Technologies Inc. Este relatório refere-se, ainda, a estratégias adotadas por estes *players* para poderem fazer face à concorrência e assim ficarem, eventualmente, habilitados a ajudarem os principais agentes do mercado na formulação e desenvolvimento de novas estratégias. Segundo Arnold (2014) uma das

estratégias em que esses e outros fornecedores de soluções *enterprise search* têm vindo a apostar é na adequação dos seus produtos ao paradigma *bigdata*.

Devemos, ainda, realçar que não foram somente os fornecedores de soluções comerciais que desenvolveram estratégias associadas ao *enterprise search*, *information retrieval* e *search engine* para conseguirem a sua adequação ao paradigma *bigdata*. As soluções *open source*, que jogam neste campo um papel relevante, também demostraram preocupação com a necessidade de adequação a este paradigma e têm vindo a desenvolver soluções com este objetivo.

De acordo com a Voyager Search (2014) todas as organizações, independentemente da sua dimensão, estão repletas de *bigdata*. A capacidade destas organizações para poderem gerir esses dados e perceberem todo o seu potencial, em muitos casos relevante, de forma a poderem garantir a sua própria sustentabilidade é um dos grandes desafios que se coloca às tecnologias associadas às principais áreas associadas à pesquisa de informação empresarial: *enterprise search*; *information retrieval*, *search engine* e obviamente *bigdata*.

2.1.5 Tecnologia

A Findwise (2014) refere que a funcionalidade associada à procura de informação empresarial é mais eficiente, trabalhando melhor a procura da informação, quando as aplicações são desenvolvidas numa abordagem *in-house* sendo portanto projetadas e implementadas de acordo com necessidades percebidas.

O objetivo da Findwise é o de ajudar as organizações a comparar a sua capacidade de poderem encontrar determinada informação, otimizando a experiência do utilizador (*Findability*), comparativamente com outras organizações, para que possam determinar o seu grau de maturidade no âmbito da procura de informação empresarial (Findwise, 2015).

Andrews e Koehler-Kruener (2015) comparam 15 fornecedores de aplicações *enterprise search* para facilitar a escolha de uma delas. No passado as soluções comerciais de *enterprise search* dominavam o mercado. No entanto, atualmente verifica-se uma tendência crescente para utilização de soluções *open source*[10].

Num estudo feito recentemente, 60% das organizações afirmaram que a pesquisa de informação é crítica para os seus negócios, mas apenas 10% estão muito satisfeitas com o desempenho do seu sistema de pesquisa de informação *in-house*. Uma possível razão para que isto aconteça pode ficar a dever-se a que apenas 20% das organizações têm mais do que uma pessoa a apoiar o sistema de pesquisa de informação. Quando os utilizadores se queixam sobre a qualidade da pesquisa, há evidência de que é feito um *upgrade* ao sistema de pesquisa, tomando como base o facto de que o processo de pesquisa não é o mais adequado, devendo ser adaptado à realidade em função das queixas apresentadas (White & Nikolov, 2013).

No Anexo 1 é mostrada uma lista de fornecedores[11], atualizada relativamente ao mês de fevereiro de 2015, de soluções para procura de informação empresarial. De acordo com a referida lista verifica-se a existência de 60 fornecedores de diversas soluções, considerando soluções comerciais a soluções *open source*.

[10] Source: Oracle Survey, *From Overload to Impact: an industry scorecard on big data business challenges*, 2012

[11] http://www.intranetfocus.com/enterprise-search/vendor-listing

2.2 Trabalho relacionado

Nesta secção são apresentados diversos trabalhos relacionados com sistemas empresariais de pesquisa, tema principal deste projeto de estágio. Pode verificar-se que a grande maioria dos trabalhos referidos se concentram entre 2010 e 2015, destacando-se assim como uma área relativamente recente e de interesse crescente.

2.2.1 Recuperação de informação

Algum do trabalho relacionado, encontrado na recolha levada a cabo, dá mais relevância a questões relacionadas com a vertente relacionada com a recuperação de informação empresarial.

Machado (2008) estuda o problema da recuperação de informação em ambientes semiestruturados, definidos como contextos, nos quais se pode verificar a existência de acesso aos conteúdos e possivelmente a estruturas descritivas dos mesmos (metadados descritivos ou estruturais). No âmbito das conclusões deste estudo merecem destaque as seguintes: verificou-se uma grande ausência de metadados; as técnicas tradicionalmente utilizadas na recuperação de informação mostraram-se suficientes; os potenciais utilizadores do sistema mostraram-se bastante agradados com a sua disponibilização e, finalmente, o sistema mostrou-se facilmente extensível aos diferentes tipos de informação estruturada e não estruturada. Como proposta de trabalho futuro, este autor, propõe algumas técnicas que poderão ser usadas para melhorar a performance e a interoperabilidade dos sistemas de pesquisa de informação em ambientes semiestruturados.

Kohn, Bry e Manta (2010) propõem um sistema de recuperação de informação, baseado em ontologias, totalmente implementado e cuidadosamente avaliado capaz de funcionar numa qualquer empresa. A mensagem chave que pode ser retirada deste estudo é a de que toda a informação que se encontre num servidor de dados de uma qualquer empresa é potencialmente útil para ajudar a melhorar a técnica de pesquisa.

Sharma (2011) desenvolve um trabalho no âmbito da procura de informação empresarial com o objetivo de aumentar a relevância dos resultados devolvidos pela procura, utilizando abordagens relacionadas com a recuperação da informação bem como o alargamento do processo e reordenação dos resultados. Atendendo aos resultados alcançados, este estudo conclui na esperança de que possa ser uma referência para a comunidade de interessados na pesquisa de informação empresarial. O autor, espera que a abordagem que utilizou neste trabalho, inclusão de repositórios de conhecimento de código aberto na pesquisa empresarial, seja um começo para aproveitar a grande quantidade de informação disponível através do conteúdo e estrutura de vários recursos disponíveis na *Web*. Espera, ainda, que este trabalho ofereça uma visão abrangente para quem se inicia no estudo e projeto de sistemas empresariais de pesquisa.

Cleverley e Burnett (2015) desenvolvem um trabalho que faz parte de um conjunto mais vasto de diversas áreas de investigação que avaliam as causas de experiências vividas na procura de informação empresarial. O estudo demonstrou que os pesquisadores empresariais (na área de negócio do petróleo e do gás) podem encontrar palavras/sugestões potencialmente úteis, com a ajuda de um filtro de navegação, para efetuar determinada pesquisa. Como potenciais áreas para trabalho futuro, os autores, identificam o teste dos resultados obtidos com uma interface de pesquisa interativa.

2.2.2 Sistemas empresariais de pesquisa

Outros autores, que desenvolveram trabalho relacionado com este projeto de estágio, focaram a sua atenção no desenvolvimento e implementação de sistemas empresariais de pesquisa, com determinadas especificidades.

Wallé (2008) projetou e implementou um sistema empresarial de pesquisa com o objetivo de utilizar uma técnica melhorada para apresentação dos resultados da pesquisa. A implementação do protótipo desenvolvido, designado BASORA, baseou-se na resposta a questões enquadradas nos seguintes tópicos: pesquisa e recuperação de informação; apresentação dos resultados da pesquisa; definição e redefinição dos termos da pesquisa; modelo cognitivo de recuperação da informação e validação dos utilizadores do sistema. Como conclusão deste estudo fica a demostração de que este sistema ajuda o utilizador a avaliar a relevância de um documento minimizando a quantidade de outros documentos que precisam de ser abertos. Como proposta de trabalho futuro fica a identificação de algumas áreas às quais pode ser estendido o estudo, utilizando o protótipo desenvolvido e implementado com pequenas modificações, designadamente: *multiple thumbnails*; algoritmos de recuperação; projeto centrado no utilizador; metadados e *enterprise parsing*.

Russel-Rose, Lamantia e Burrell (2011) apresentaram um trabalho em que estendem a pesquisa de informação ao contexto empresarial, focando a sua atenção nas necessidades e no comportamento dos utilizadores da informação, considerando vários cenários e tipos de empresa. Para isto apresentam uma taxonomia de "modo de pesquisa" e discutem algumas implicações que, em sua opinião, podem ajudar a melhorar o desempenho da plataforma e ferramentas de pesquisa utilizadas. Concluem, os autores, que, para que se possam projetar melhores experiências de pesquisa, devem ser compreendidas as complexidades do processo de interação do binómio humano-informação que se verifica em qualquer pesquisa de informação empresarial ou não. É sugerido, pelos autores, que este trabalho seja complementado com a inclusão de novas iterações no ciclo "propor-classificar-refinar" com recurso a um conjunto de dados independentes.

Escada (2012) desenvolveu um projeto, em ambiente empresarial, em que teve como objetivo implementar um sistema de pesquisa de informação, numa determinada organização, tendo em atenção a necessidade de garantir a proteção da informação relativamente a acessos não autorizados. Os objetivos, definidos inicialmente, foram maioritariamente cumpridos, tendo sido criado um sistema que permite pesquisar conteúdo partilhado numa rede e disponibilizá-lo para que com uma ferramenta externa se proceda à análise e proteção do seu conteúdo. Sugere o autor que, complementarmente, ao seu estudo sejam feitos esforços de melhoria nas seguintes áreas: suporte para outros tipos de repositórios de dados; relatórios e notificações; *logging* de atividades e interface gráfica. Verifica-se assim, que há ainda muito trabalho que pode ser desenvolvido no âmbito desta ferramenta.

Reinhart (2013) procura forma de responder à necessidade, sentida, de localizar especialistas em determinada área e de que forma pode ser melhorado este tipo de pesquisa utilizando técnicas enquadradas na pesquisa de informação empresarial. Este autor conclui demonstrando a eficácia das técnicas empresariais de pesquisa de informação para atingir o objetivo pretendido e declarado no início do seu estudo, assim como a implementação de melhores práticas e medidas de avaliação no sentido de localizar especialistas em determinadas áreas. Também não é esquecida a necessidade de bons repositórios de informação para que os resultados alcançados tenham qualidade. O autor entende que o sistema desenvolvido pode ser utilizado noutros domínios que necessitem de localizar

especialistas em determinada área, não deixando de continuar a trabalhar na melhoria da apresentação dos resultados obtidos.

Segundo Wilhelmsson e Eriksson (2013), para que a criação de um roteiro estratégico para uma determinada organização, seja facilitado, deve ser definido o estado da organização, tanto em termos presentes, quanto em termos futuros. Assim, os autores, desenvolvem um modelo que permite avaliar o nível de maturidade de um sistema empresarial de pesquisa de informação. Os autores propõem, como trabalho futuro, que o modelo desenvolvido seja devidamente testado. A realização de vários testes ajudará a obter uma visão mais precisa dos prós e contras do modelo, aumentando a confiabilidade, validade e generalização dos seus resultados.

Wu, Turpin, Thom, Scholer e Wilkinson (2014), propõem e avaliam um modelo de custo/benefício para o fornecedor do serviço quando investe na melhoria do *ranking* dos resultados devolvidos pela pesquisa que é feita. A abordagem de análise de custo-benefício proposta é complementar às técnicas mais tradicionais que são usadas para avaliar a eficácia dos sistemas empresariais de pesquisa. Segundo os autores, este estudo demonstra que num cenário empresarial, os dados qualitativos com base nas preferências do utilizador, por si só, não contam toda a história sobre se a introdução de capacidades de pesquisa avançadas trará benefícios para a organização. Os autores entendem que seria interessante aplicar uma análise semelhante, àquela que é apresentada no seu estudo, num contexto mais amplo, tal como bibliotecas digitais publicamente acessíveis.

Stenmark, Gårdelöv e Larsson (2015) apresentam um estudo exploratório em que investigam a pesquisa de informação empresarial na perspetiva da governação das tecnologias de informação e comunicação. Os autores concluem, com este trabalho, que a capacidade de encontrar determinada informação e a satisfação do utilizador são afetadas pela forma como as organizações implementam a sua política de pesquisa de informação empresarial. Os autores recomendam que as organizações, nomeadamente as grandes, tomem o desenvolvimento e implementação de sistemas empresariais de pesquisa como uma atividade estratégica.

2.2.3 Funcionalidades de pesquisa empresarial avançada

Ainda no âmbito do trabalho relacionado com o tema principal deste projeto de estágio, alguns dos estudos identificados, manifestam preocupação com a apreciação de funcionalidades de pesquisa empresarial avançada.

Alves (2010), em contexto empresarial, desenvolveu um projeto com o objetivo de integrar numa ferramenta já existente, ou desenvolver uma nova ferramenta, para disponibilizar funcionalidades de pesquisa empresarial avançada baseada nas últimas tecnologias de pesquisa existentes no mercado. Como conclusão devemos referir que este estudo atingiu os objetivos previamente definidos para desenvolvimento de um *front-end* integrado numa ferramenta existente. Este autor entende que, na área de investigação relacionada com os sistemas empresariais de pesquisa, há ainda muito a fazer, considerando que só recentemente a pesquisa de informação começou a ser desenvolvida no âmbito empresarial.

Sanaka (2010) explica o procedimento para extrair e organizar notícias e dados a partir de *websites* e desenvolver uma interface com capacidade para pesquisa de facetas. Esta ferramenta pode ser de grande utilidade para analisar as várias atividades desenvolvidas por qualquer organização. O autor, refere como próximos passos a possibilidade de melhorar o servidor de pesquisa otimizando o seu desempenho e fornecendo uma interface

administrativa, assim como pensar numa abordagem que permita aumentar o número de facetas ampliando o campo de aplicação da ferramenta tornando-a mais escalável.

Moura (2012) desenvolveu, igualmente, um trabalho, em contexto empresarial, associado à pesquisa de informação mas focado na necessidade de garantir a qualidade dos resultados obtidos, o desempenho e a facilidade de utilização do sistema. Este estudo permitiu complementar um sistema existente que, no entanto, padecia de algumas limitações designadamente: ao nível dos resultados provenientes da interação do utilizador com o motor de pesquisa da organização em que o mesmo foi desenvolvido e de análise dessa mesma informação. Tornou-se assim possível tratar informação que de outra maneira seria de difícil compreensão, visualização e análise analítica. Relativamente a trabalho futuro, o autor identificou as seguintes tarefas: implementar um sistema de alerta para variações abrutas de métricas que possa ser configurável relativamente às métricas a monitorizar e valores para os quais devem ser emitidos os alertas e ter mais informação pré-calculada, como por exemplo: o *hit rate*; o *bounce rate* e o *refinement rate* de forma a minimizar o tempo de resposta ao pedido de informação.

Bisson, Patel e Pasupathy (2012), neste trabalho, os autores, discutem os desafios que se colocam na implementação de um *crawler* para pesquisar ficheiros apresentando o projeto de dois *crawlers* para sistemas de ficheiros: um deles utiliza o standard POSIX da API para sistemas de ficheiros, sendo feito um controlo cuidadoso das quantidades de memória e CPU utilizadas; o outro introduz modificações e uma nova API designada *snapdiff*, para rapidamente detetar modificações. Os autores, como trabalho futuro, pretendem adicionar funcionalidade para monitorar dinamicamente o uso da CPU a partir do *crawler*, o que permitirá que o número de *threads* ativas seja ajustado em tempo de execução. Acreditam, os autores, que há otimizações que podem ser feitas para melhorar o paralelismo.

Liu, Fang, Chen e Wang (2012) propõem um novo método de pesquisa de informação empresarial centrada na entidade (este tipo de abordagem decide se uma menção pertence a um conjunto de menções da mesma entidade utilizando atributos de outras menções dessa mesma entidade (González, 2014)), formalizando uma *framework* adaptada a este objetivo. Os autores concluem afirmando que os resultados experimentais demonstram que os métodos propostos de hierarquização das entidades podem ajudar a recuperar informação daquelas entidades com um elevado índice de qualidade. Os mesmos autores publicam novo estudo (2014) em que apresentam mais detalhadamente os resultados experimentais da *framework* proposta chegando às mesmas conclusões apresentadas no estudo anterior. Os autores acreditam que este trabalho propõe várias pistas para trabalho futuro. Em primeiro lugar, seria interessante, para alavancar os métodos de apresentação dos resultados, utilizar outros tipos de relações que ajudassem a melhorar o desempenho do sistema. Em segundo lugar, propõem que sejam estudadas formas alternativas de combinação de diferentes tipos de relações. Em terceiro lugar, sugerem que seja estudada a forma de utilizar as entidades relacionadas para agregar resultados de pesquisa. Finalmente, pensam os autores que seria interessante avaliar a eficácia dos métodos que propõem em domínios de procura de informação diferentes daquele em que trabalharam.

Woo (2013) propõe um modelo para integração de dados a partir de múltiplas fontes heterogéneas, ilustrando o projeto de possível sistema e abordagem utilizada para implementação do modelo.

Grant e Schymik (2014) manifestam no seu trabalho a opinião de que a "*work system theory*" proposta por Alter (2013) poderá fornecer uma nova perspetiva, segundo a qual poderá ser eliminada grande parte da insatisfação dos utilizadores que pesquisam informação empresarial. Esta teoria tira a ênfase da tecnologia na análise de um sistema de trabalho e

enfatiza os clientes, produtos e serviços, atividades e processos, informações e elementos participantes. Os autores sugerem que os investigadores devem procurar melhorar e testar o modelo proposto em vários outros cenários.

2.2.4 Plataformas open source

Finalmente, destacamos um conjunto de trabalhos relacionados com o tema deste projeto de estágio, que estão focados na utilização de plataformas *open source*.

Molková (2011) desenvolveu um estudo sobre a utilização da plataforma Lucene para indexação de um elevado número de ficheiros. Conclui, este estudo, que aquela plataforma disponibiliza um importante conjunto de funcionalidades que facilitam a recuperação de informação empresarial. No entanto, não é um produto independente pronto a usar, não contendo um *parser* , filtragem de documentos ou uma interface de pesquisa. Como trabalho futuro é proposto que sejam adotadas medidas para melhorar o desempenho da plataforma Lucene.

Alhabashneh, Iqbal, Shah, Amin e James (2011) apresentam uma *framework,* baseada em tecnologias *open source,* com o objetivo de melhorar a pesquisa de informação empresarial, prevendo que esta *framework* poderá beneficiar várias empresas, melhorando a sua produtividade e cumprindo as necessidades de informação de forma eficaz. Como trabalho complementar, propõem os autores, que seja feita uma aposta num maior desenvolvimento da *framework* proposta, realizando-se várias experiências com conjuntos de dados teste para avaliar a sua eficácia. Deverão, igualmente, ser utilizadas diferentes métricas para medir a precisão dos resultados obtidos, a eficiência da execução e a satisfação do utilizador.

Khabsa, Carman, Choudhury e Giles (2012) relatam a sua experiência no desenvolvimento de um sistema empresarial de pesquisa de informação (*YouSeer*), tomando como abordagem a pesquisa por segmentos específicos na *Web* ou nas organizações. Este sistema foi desenvolvido com recurso a ferramentas *open source* (tanto no que diz respeito ao *crawler*, quanto no que diz respeito ao indexador). Os autores puderam concluir que a sua experiência mostra que o sistema desenvolvido pode ser mais efetivo do que outros sistemas de pesquisa de informação empresarial, *open source*, em certos cenários. Futuramente, os autores, planeiam introduzir módulos para paralelizar o processamento e tirar proveito do paradigma *MapReduce*, assim como, investigar sobre modelos de segurança que protejam os dados de acessos por utilizadores não autorizados.

Pettersson e Pettersson (2013) estão preocupados em saber da viabilidade de, integrando modernas tecnologias de pesquisa, ser possível a procura de produtos através da implementação de um protótipo empresarial de pesquisa baseado na plataforma Lucene.NET. Os autores, concluem pela viabilidade da execução do sistema pretendido recorrendo à plataforma Lucene.NET, acreditando que sendo dedicado algum tempo adicional para ajustar a configuração do Lucene, será certamente possível reduzir ainda mais os tempos de resposta gastos nas operações de pesquisa.

Ravikumar (2014) desenvolveu um estudo com o objetivo de avaliar os conceitos básicos da plataforma *open source* Apache Solr projetada especificamente para indexação de documentos e execução de pesquisa de informação empresarial.

O'Riordan (2014) apresenta um trabalho em que explora o espaço das tecnologias de pesquisa aberta. Aproveitando, bibliotecas de software livre para construir motores de pesquisa de informação disponíveis há algum tempo e que permitem construir soluções de pesquisa em alternativa aos sistemas proprietários, através da construção de *crawlers* à medida das necessidades dos potenciais utilizadores permitindo a recolha de informação a partir de

várias fontes. No entanto coloca-se o problema de encontrar a melhor forma de satisfazer as expetativas dos utilizadores quanto à eficácia dos resultados obtidos, considerando que não estão disponíveis os recursos dos grandes fornecedores de sistemas empresariais de pesquisa. No entanto, os autores consideram que estas reservas merecem ser desafiadas para que sejam encontradas alternativas *open source* que sejam de facto opção relativamente aos sistemas proprietários. O autor, considera que uma área promissora de desenvolvimento adicional seria a de tornar o trabalho com padrões abertos mais acessível a desenvolvedores e até mesmo, mais acessível, a não programadores.

Valentín-Rodríguez, Ojeda-Castro, Alanís-González, Márquez-Martínez e Curbelo-Ruiz (2015) determinam, no seu estudo, alguns dos fatores que podem influenciar a adoção de software *open source*. Os autores concluem que o custo é um factor determinante a considerar quando se pretendem adotar tecnologias *open source*.

2.3 Tecnologias utilizadas

O desenvolvimento do protótipo do sistema CYCLOPS exigiu a utilização de diversas tecnologias, algumas utilizadas para o desenvolvimento e implementação de alguns dos módulos utilizados, outras para implementar soluções já existentes e devidamente testadas em trabalhos relacionados com aquele que aqui desenvolvemos, para interligação dos referidos módulos e desenvolvimento da interface *Web*. Nas subsecções seguintes é apresentada uma pequena resenha de cada uma das tecnologias utilizadas.

Outras tecnologias poderiam ter sido utilizadas. No entanto a opção pela utilização de tecnologias *open source* limitou o espectro a este tipo de soluções. Assim, foram analisadas as seguintes plataformas de pesquisa empresarial: Search Daimon: Apache Solr e ElasticSearch. Sobre estas plataforma Almeida (2014) faz uma análise bastante completa, comparando as características das três plataformas. Tendo em atenção essa análise comparativa: a capacidade para permissões, o modo de operação próximo do tempo real e ser uma tecnologia que vem sendo permanetmente atualizada desde 2004 e existir uma vasta literatura de suporte, a opção para desenvolvimento do protótipo do sistema CYCLOPS recaiu sobre a plataforma Apache Solr.

Tendo sido tomada esta opção, também reforçada pelos estudos de Middleton e Baeza-Yates (2011; 2007), numa lógica de implementação *enterprise-wide* (utilizar diferentes módulos do mesmo fornecedor), as restantes tecnologias que foi necessário utilizar, designadamente para fazer a gestão de mensagens entre módulos e extrair metaddos dos documentos procurados, optamos por utilizar sistemas *open source* disponibilizados pela Apache Software Foundation.

2.3.1 Linguagem de programação Java

Para desenvolvimento de alguns dos módulos, que constituem o protótipo do sistema CYCLOPS, designadamente: *crawler*; *parser* e *indexer*, optamos por utilizar a linguagem de programação Java. Esta linguagem disponibiliza um largo manancial de recursos tecnológicos para responder às mais diversas necessidades que se colocam a quem pretender desenvolver aplicações em ambiente distribuído, com um determinado nível de segurança e facilidade de manutenção. Estes são alguns dos requisitos que sustentam a opção pela tecnologia Java (Martins, 2009).

2.3.2 HyperText Markup Language - HTML

Para desenvolver a interface *Web* utilizada neste protótipo, recorremos a diversas tecnologias, entre elas destacamos a linguagem de marcação HTML5. Esta tecnologia procura organizar o caos, que por vezes se sente e vive no decorrer de um processo de desenvolvimento de interfaces *Web*, codificando práticas comuns, abraçando o que já está implementado em navegadores e documentando de que forma os dispositivos que consomem páginas *Web* devem lidar com a nossa capacidade de marcação (Powel, 2010).

2.3.3 Cascading Style Sheets - CSS

As interfaces *Web* procuram dar qualidade à apresentação dos diversos elementos e atributos que compõem a referida interface, proporcionando assim uma clara distinção entre a estrutura fornecida por marcação (HTML) e o layout proporcionado pela utilização de uma folha de estilo escrita em Cascading Style Sheets (CSS). A divisão nítida de funções entre marcação e estilo pode fornecer numerosas vantagens à produção, manutenção, e até mesmo ao desempenho da interface (Powel, 2010).

2.3.4 Linguagem de programação JavaScript

Para além da linguagem HTML utilizada para estruturar e definir semanticamente os elementos presentes na interface *Web*, houve necessidade de recorrer à tecnologia CSS para garantir que o *layout* desta interface fosse mais adequado aos objetivos do protótipo. No entanto, estas duas tecnologias (HTML e CSS) não são suficientes para garantir uma interface *Web* que interaja adequadamente com os seus utilizadores. Para conseguir uma boa interação entre a interface *Web* e os seus utilizadores foi utilizada a tecnologia JavaScript (Douglas, 2014).

2.3.5 Apache Tika

O Apache Tika, no essencial, é uma biblioteca que disponibiliza facilidades para identificar o tipo de um determinado documento e a extração de conteúdo a partir de vários formatos. Assim, a utilização desta framework permitirá desenvolver um mecanismo de extração de texto estruturado bem como metadados de diferentes tipos de documentos [12]

2.3.6 Apache Kafka

Atualmente existe informação que está a ser gerada de forma contínua pelos mais diversos tipos de sistemas (negócios, sociais, ou qualquer outro tipo). Esta informação deve ser confiável e rapidamente encaminhada para os seus potenciais consumidores. Assim, é necessário um mecanismo que facilite a integração de informação entre produtores e consumidores. Para dar resposta a esta necessidade existe o Apache Kafka (Garg, 2015).

O Apache Kafka[13] é um sistema de mensagens *publisher-subscriver* distribuído, desenhado para alta performance, criado inicialmente pela rede social de negócios LinkedIn[14]. Este sistema foi posteriormente disponibilizado ao público em geral, em 2012, através de uma licença *open source*, passando a fazer parta da Apache Software Foundation. A rede social de negócios LinkedIn, quando desenvolveu o Kafka, focou-se na criação de um sistema de alto

[12] http://www.w3ii.com/pt/tika/tika_quick_guide.html
[13] http://kafka.apache.org/
[14] https://pt.linkedin.com/

desempenho capaz de suportar fluxos de grandes volumes de dados, assim como suportar a distribuição e processamento desses fluxos em tempo-real (Andrade, 2015; Esteso, 2015).

O sistema de gestão de mensagens Apache Kafka distingue-se dos demais nalguns aspetos relevantes: está projetado para escalar; oferece alto rendimento tanto na publicação quanto na subscrição, suportando mais do que uma subscrição simultaneamente; tem balanceamento automático em caso de falha e as mensagens são persistentes no disco (Esteso, 2015).

2.3.7 Apache Solr

O Apache Solr é uma plataforma de pesquisa empresarial, igualmente, mantida pela Apache Software Foundation que tem como objetivo procurar informação através de um motor de pesquisa aberto baseado no Lucene.

Para além de todas as funcionalidades inerentes ao Lucene o Apache Solr permite indexar ficheiros XML de forma configurável permitindo definir para cada esquema quais os campos a indexar. Esta plataforma não dispõe de serviço de *crawling* tendo de ser feito caso a caso um módulo de indexação que envie os documentos desejados para o sistema. Tem a desvantagem de não indexar documentos que não estejam em XML obrigando a uma conversão prévia (Machado, 2008).

2.3.8 Apache ZooKeeper

O Apache ZooKeeper, igualmente, mantido pela Apache Software Foundation é um sistema de gestão e coordenação de vários serviços num sistema distribuído em ambientes *cluster*. O Apache ZooKeeper resolve a dificuldade da gestão e coordenação em ambiente distribuído com a sua arquitetura simples e API, permitindo que quem desenvolve sistemas se concentre unicamente na lógica da aplicação sem se preocupar com a sua natureza distribuída[15].

O sistema Apache ZooKeeper foi originalmente desenvolvido pela Yahoo! para facilitar o acesso às suas aplicações de uma forma simples e robusta (Haloi, 2015). Mais tarde, o tornou-se um padrão usado, entre outros, pelo sistema de gestão de mensagens Apache Kafka.

2.3.9 Apache Maven

Para realização dos testes e verificação do sistema foi utilizada a *framework open source* Apache Maven, igualmente mantida pela Apache software Foundation. Esta *framework*, em ambientes que envolvam várias equipas de desenvolvimento, facilita uma forma de trabalhar, cumprindo regras, num tempo muito curto. A maioria das configurações são simples e reutilizáveis, simplificando algumas das tarefas do desenvolvedor de sistemas, designadamente: criação de relatórios; verificações e construção de configurações para testes e automatização.

2.3.10 Framework JUnit

Ainda no âmbito dos testes e verificação do sistema recorremos à *framework* JUnit. Esta *framework* é uma das ferramentas utilizadas pelo Apache Maven para automatização de casos de teste. Os relatórios de execução destes casos de teste são produzidos com recurso ao *plugin* surefire, um dos *plugins* mais comuns do Apache Maven.

[15] http://www.tutorialspoint.com/zookeeper/zookeeper_overview.htm

[Esta página foi intencionalmente deixada em branco]

Capítulo 3
Objetivos e Metodologia

Neste capítulo é apresentado de forma detalhada o problema, identificado no Capítulo 1, ao qual deve ser dada resposta através de uma solução adequada. São igualmente identificados os objetivos que se pretendem atingir, com a realização deste projeto e é apresentada a abordagem metodológica utilizada, assim como as razões que levaram às opções tomadas. É, igualmente, apresentado o planeamento que orientou o desenvolvimento e implementação do protótipo do sistema CYCLOPS assim como a identificação e avaliação dos riscos associados a cada um dos *Sprints* relativos a este planeamento e os mecanismos de controlo adequados para minimizar a exposição aos riscos identificados tomando como referência a metodologia analítica, *Failure Mode and effect Analysis* (FMEA), que dsignaremos análise modal de falhas.

Como já foi referido anteriormente, o sistema CYCLOPS, é um projeto de I&D promovido pela CRITICAL Software, SA cujo âmbito se enquadra no desenvolvimento de um protótipo de um sistema de informação empresarial que facilite o acesso à informação armazenada nos seus repositórios corporativos. Conforme referido no Capítulo 1, por questões de tempo, vamos limitar a pesquisa a dois destes repositórios: (1) o repositório *SVN*, este repositório é um sistema *open source* para controlo de versões, gerindo as alterações feitas ao longo do tempo em ficheiros e/ou diretorias; (2) e o repositório *AsknowledgeDb*, uma base de dados que contém os dados relativos a todos os colaboradores da CRITICAL Software, SA.

3.1 Clarificação do problema

Conforme referido no Capítulo 1 e com base no trabalho relacionado com o âmbito deste projeto, independentemente dos avanços que têm sido feitos no domínio dos sistemas empresarias de pesquisa, ainda subsistem várias questões que são merecedoras de investigação específica (Alves, 2010; Escada, 2012; Grant & Schymik, 2014; Molková, 2011; Moura, 2012; Pettersson & Pettersson, 2013; Wilhelmsson & Eriksson, 2013). Estas questões não devem, como é óbvio, descurar os custos que estão sempre associados ao desenvolvimento e implementação de qualquer tipo de sistema e deste em particular (Valentín-Rodríguez et al., 2015; Wu et al., 2014).

Para além dos aspetos referidos no parágrafo anterior de caráter geral, existem vários outros de cariz técnico que também foram tidos em atenção. Assim, Wallé (2008) ao identificar algumas áreas às quais possa ser estendido o âmbito do seu estudo refere, entre elas, os metadados e o *enterprise parsing*. Neste projeto também é considerado que estas áreas são relevantes no desenvolvimento e implementação de um sistema empresarial de pesquisa e que, as mesmas, vão de encontro às necessidades da CRITICAL Software, SA.

Deve, igualmente, ser destacado que vários estudos focam como relevante o aspeto da utilização de tecnologias *open source* no desenvolvimento de sistemas empresariais de pesquisa. O'Riordam (2014), conforme já referido no capítulo anterior, apresenta um estudo que explora as potencialidades das tecnologias *open source*, considerando que devem ser encontradas alternativas, destas tecnologias, aos sistemas comerciais, justificando, este interesse, que esta área deve ser merecedora de desenvolvimento adicional. Por seu lado Valentín-Rodriguez *et al* (2015), igualmente referidos no capítulo anterior, desenvolvem um estudo em que determinam os fatores que podem influenciar a adoção de software *open source* no âmbito do desenvolvimento de sistemas empresariais de pesquisa.

Outros estudos devem ainda ser considerados, conforme referido no capítulo anterior, no âmbito da utilização de tecnologias *open source* aplicadas ao desenvolvimento de sistemas empresariais de pesquisa. Molková (2011) disserta sobre a utilização da plataforma Apache Lucene para indexação de um elevado número de ficheiros, propondo que sejam adotadas medidas que tenham por objetivo melhorar o seu desempenho. Também Pettersson e Pettersson (2013) utilizaram a plataforma Lucene.NET no seu estudo relacionado com a pesquisa de informação empresarial. Alhabashneh et al (2011) e Khabsa et al (2012) baseiam o trabalho que desenvolveram na utilização de *frameworks*, suportadas em tecnologias *open source*. Finalmente Ravikumar (2014) avalia a plataforma Apache Solr para indexação de documentos e pesquisa de informação empresarial.

Tendo em atenção os trabalhos referidos, assim como as necessidades da CRITICAL Software, SA, entendemos apropriado utilizar, sempre que possível, no desenvolvimento e implementação deste projeto, tecnologias *open source,* com destaque, entre outras, para a plataforma Apache Solr.

Outra questão relevante identificada pela CRITICAL Software, SA está relacionada com a segurança do sistema. Esta questão está igualmente sinalizada como uma preocupação a ter em atenção no desenvolvimento de sistemas empresariais de pesquisa (Khabsa et al., 2012). Igualmente importante é a questão relacionada com a relevância dos resultados devolvidos pela procura, sendo esta questão igualmente sinalizada em diversos trabalhos relacionados com destaque para Wallé (2008), que conclui no seu estudo que o sistema que propõe pode ajudar o utilizador a avaliar a relevância dos resultados e Sharma (2011) cujo trabalho tem por objetivo aumentar a relevância dos resultados obtidos pela pesquisa.

Assim, face ao que atrás é exposto, pode ser clarificado o desafio, identificado no âmbito do sistema CYCLOPS, que carece de uma resposta adequada. Assim, o desafio que se coloca é o seguinte:

desenvolver e implementar um protótipo do sistema CYCLOPS, focado na procura de documentos nos repositórios corporativos: *SVN* e *AsknowledgeDb*, tendo em atenção: o acesso/permissões; a relevância dos dados; os tempos de resposta e a extensibilidade do sistema.

Para responder ao deafio, apresentado acima, deve ser garantido o cuprimento dos seguintes objetivos:

- pesquisa de informação em repositórios corporativos;
- pesquisa e apresentação dos resultados através de uma interface *Web*;
- garantia de segurança e controlo de acesso; e,
- operacionalidade do sistema implementado.

O cumprimento destes objetivos pressupõe que, previamente, sejam encontradas respostas para algumas questões que lhe estão claramente associadas, designadamente:

1. que requisitos, não funcionais e funcionais devem ser considerados?
2. que arquitetura deve ser utilizada para responder de forma adequada ao pretendido?
3. que tecnologias são adequadas à implementação dos requisitos identificados?
4. a arquitetura e as tecnologias propostas garantem a extensibilidade do sistema?
5. a arquitetura e as tecnologias propostas garantem que as questões relativas à qualidade e segurança da informação são acauteladas?

Para obter as respostas necessárias, às questões colocadas anteriormente, e considerando que a área de desenvolvimento e implementação de sistemas empresariais de pesquisa ainda não se encontra devidamente consolidada foi feito um estudo exploratório, baseado em textos existentes, sobre a temática associada aos sistemas de procura de informação empresarial e que poderiam ser elucidativos sobre cada uma daquelas questões. Para além deste estudo exploratório, para identificar os requisitos não funcionais e funcionais, o estagiário por estar integrado numa equipa de I&D, recorreu a uma abordagem etnográfica. Para complementar esta abordagem foi elaborado o questionário apresentado no Anexo 2 para sentir a opinião dos colaboradores das áreas técnicas da iTGROW e da CRITICAL Software, SA. Finalmente, foi entendido que a opção mais adequada para desenvolvimento e implementação do protótipo CYCLOPS seria a utilização de metodologias ágeis.

3.2 Abordagem metodológica

De seguida tecemos algumas considerações sobre a abordagem metodológica utilizada no estudo, desenvolvimento e implementação deste projeto, tanto no que respeita à fase conceptual, quanto no que respeita à fase de desenvolvimento e implementação do presente protótipo.

3.2.1 Fase conceptual

A fase conceptual é a fase que permite tomar contacto com o tema do projeto, estudá-lo e percebê-lo para poder detalhar o problema identificado, os objetivos e as questões que devem ser respondidas, estabelecendo-se desta forma os fundamentos do estudo (Fortin, Côté, & Filion, 2006).

Nesta fase o estagiário teve necessidade de se familiarizar com o âmbito do sistema CYCLOPS. Para isso, o referido estagiário realizou um estudo preliminar que envolveu, fundamentalmente, o levantamenteo bibliográfico relacionado com o tema, tanto ao nível de conceitos, quanto ao nível de trabalho relacionado. Pode assim dizer-se que se começou por desenvolver um estudo exploratório, complementado com uma abordagem etnográfica.

A identificação e especificação de requisitos consiste em recolher, entender e documentar os requisitos considerados relevantes no âmbito de desenvolvimento e implementação do protótipo associado ao sistema CYCLOPS. Esta identificação e especificação pode ser feita segundo diferentes abordagens. Tendo em atenção o envolvimento do estagiário, com a estrutura organizacional, será utilizada uma abordagem etnográfica, complementada com respostas recolhidas junto de colaboradores das áreas técnicas da iTGROW e da CRITICAL Software, SA que foram respondentes ao questionário, apresentado no Anexo 2.

Um sistema de pesquisa de informação empresarial deve ser visto como um componente crítico, fazendo parte da infraestrutura que suporta cada departamento da empresa (áreas produtivas ou áreas de suporte), tendo obviamente grande impacto na produtividade da organização, designadamente sobre a sua eficiência operacional e sobre informação relativamente à melhor utilização dos seus ativos (Leher, 2014). Por outro lado Stenmark *et*

al (2015) recomendam que as organizações olhem para os sistemas empresariais de pesquisa como um ativo estratégico. De acordo com Croft *et al* (2015), as principais questões que se colocam no desenvolvimento de um sistema empresarial de pesquisa de informação colocam-se ao nível das seguintes vertentes: *information retrieval* e *search engine*, ver Figura 4.

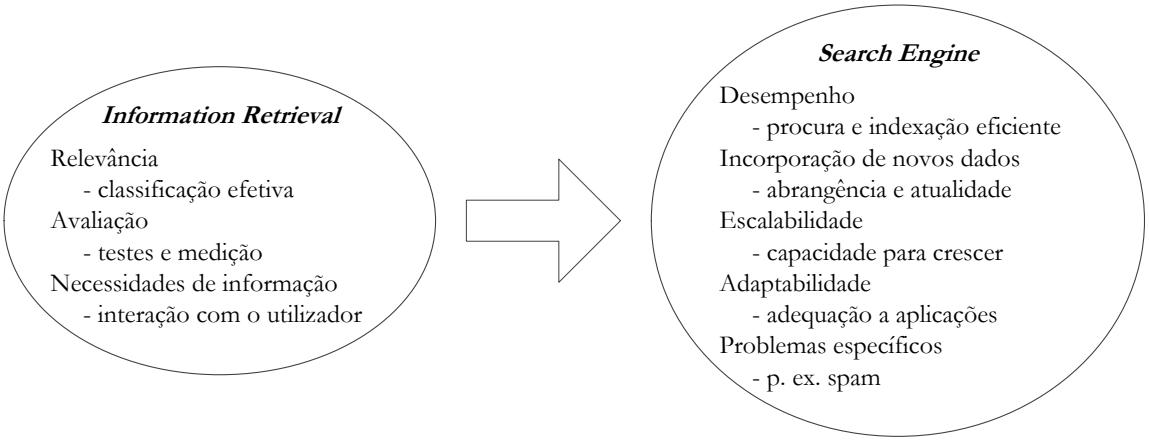


Figura 4 - Desenvolvimento de um sistema de pesquisa de informação (Croft et al., 2015)

A Figura 4, para além de levantar um um importante conjunto de questões relacionadas com information retrieval e dearch engine, também permite identificar um conjunto considerável de atributos de qualidade, que devem ser sempre tidos em atenção quando se identificam e especificam requisitos.

3.2.2 Fase metodológica

Segundo Fortin *et al.* (2006) passa-se à fase metodológica, aquela que operacionaliza o estudo, depois de se ter formulado o problema, detalhado os objetivos e identificado as questões a que é necessário responder para obter uma solução que dê resposta ao problema. Nesta fase optámos pelo desenvolvimento ágil de software, tendo em atenção que os seus princípios valorizam mais: os indivíduos e as interações do que os processos e as ferramentas; o software funcional do que documentação abrangente; a colaboração com o cliente do que a negociação contratual e a resposta à mudança do que seguir um plano[16]. Segundo Mahanti (2006) vários estudos têm mostrado que as metodologias ágeis são uma maneira eficiente de produzir software com vantagens significativas em custos de produção, entrada em operação, complexidade e melhoria da qualidade relativamente às metodologias tradicionais.

Alterações de última hora ou a introdução de novos requisitos, normalmente provocam grande impacto num projeto em desenvolvimento. Muitas vezes, quem desenvolve os projetos tem necessidade de reformular inteiramente o projeto. Assim, o desafio que se coloca a um projeto, no âmbito das tecnologias de informação e comunicação, consiste em saber como satisfazer as necessidades em mudança de forma rápida, precisa e ágil, ou seja, estes projetos devem ser ágeis (Simon, 2010). A maioria das organizações que adotam métodos ágeis identificaram como benefício desta utilização a redução do tempo necessário para colocar os sistemas em operação, assim como ganhos de produtividade entre 30 a 40 por cento, juntamente com o aumento da satisfação dos parceiros de negócio e dos colaboradores na área das tecnologias de informação e comunicação (Simon, 2010).

Conforme referido anteriormente, tendo em atenção o envolvimento do estagiário na equipa de I&D da área *Data & Business Analytics* da iTGROW, para identificação e especificação dos

[16] http://agilemanifesto.org/

requisitos foi utilizada uma abordagem etnográfica complementada com a elaboração de um questionário, Anexo 2. Segundo Hughes, King, Rodden, e Andersen (1995) uma das vantagens da metodologia etnográfica consiste na capacidade associada a este tipo de abordagem para tornar visíveis aspetos do mundo real.

Para uma adequada especificação dos requisitos, tanto não funcionais quanto funcionais, deve ser dada atenção a todos os potenciais utilizadores de informação, que de acordo com Leher (2014) serão todas as áreas, áreas de suporte ao negócio e áreas técnicas, ver Figura 5.

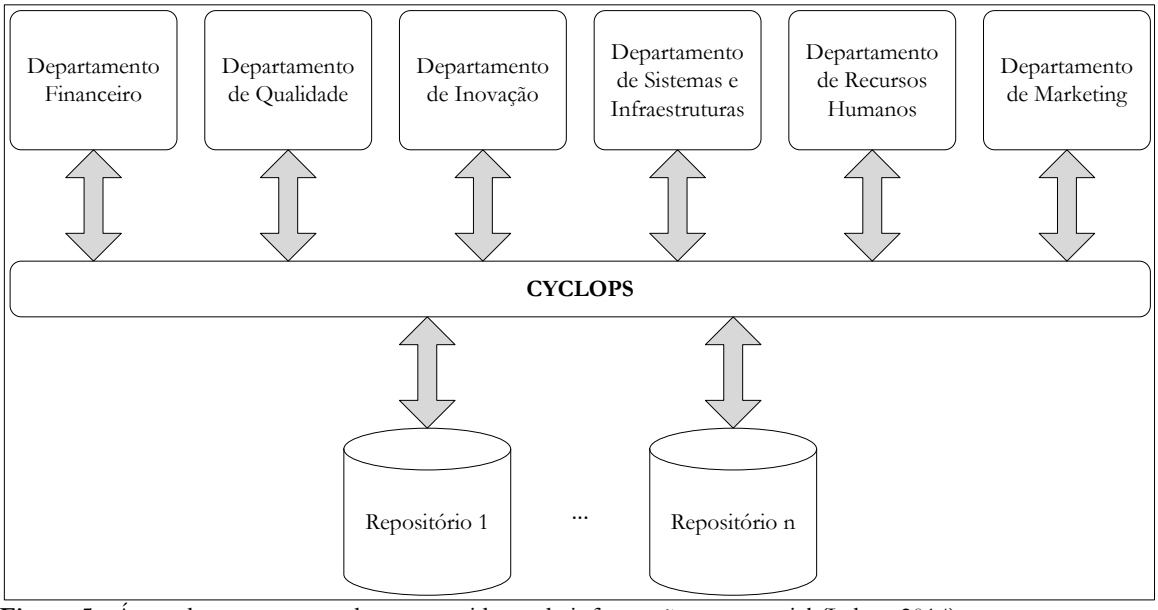


Figura 5 - Áreas de suporte, grandes consumidores de informação empresarial (Leher, 2014)

No entanto, atendendo à especificidade da empresa em que se desenvolveu o projeto de estágio, relativamente à identificação dos requisitos funcionais entendemos ser importante elaborar um questionário, desenvolvido utilizando a ferramenta *Google Forms*[17] Anexo 2, divulgado por todos os colaboradores associados às áreas técnicas da iTGROW e da CRITICAL Software, SA. Para garantir que este questionário somente seria respondido uma vez, por cada um dos respondentes, o mesmo só pôde ser preenchido por utilizadores com conta no Google.

O estagiário realizou diversas reuniões com o orientador por parte da CRITICAL Software, SA e outros colaboradores da referida empresa em que expunha o trabalho desenvolvido e obtinha o respetivo *feedback*. Assim foi possível ir avaliando se o trabalho feito se enquadrava com o pretendido pela empresa. Portanto, cada uma das etapas do projeto foi discutida e validada com o cliente, avançando-se para a etapa seguinte do projeto somente após validação da etapa anterior por parte do mesmo.

Estas reuniões serviram, essencialmente, para validar o trabalho desenvolvido na avaliação do estado da arte, que se verifica, no âmbito da pesquisa de informação empresarial, assim como para valiadação do levantamento de requisitos e implementação do protótipo associado ao sistema CYCLOPS.

[17] https://www.google.com/forms/about/

3.3 Metodologia de desenvolvimento

Considerando que o projeto em causa lida diretamente com a necessidade das pessoas, procurarem informação organizacional necessária à sua atividade profissional, entendeu-se que a metodologia a adotar deveria pressupor que estas têm um papel fundamental no desenvolvimento e implementação deste projeto, valorizando-se a entrega de um produto adequado às necessidades efetivas dos utilizadores (Tomás, 2009).

Os princípios que regem as metodologias ágeis foram publicados sob a forma de manifesto em 2001[18]. Neste manifesto são definidos doze princípios a partir dos quais têm sido criados modelos ágeis de processo. Quando se utilizam estes modelos, não raras vezes, o utilizador é chamado a ter algum papel no processo. Desde logo será chamado a colaborar no levantamento de requisitos. Na base da opção por esta metodologia também teve relevância a constatação de que normalmente um projeto não é totalmente previsível, devendo aceitar-se que os requisitos possam mudar durante o seu desenvolvimento, sendo aconselhável que a metodologia adotada possa acomodar estas mudanças (Tomás, 2009).

As metodologias ágeis, para desenvolvimento de software, surgiram como reação aos métodos de desenvolvimento tradicionais, normalmente: pesados, lentos e caros (Barbosa, Azevedo, Pereira, Campos, & Santos, 2007). As metodologias ágeis, normalmente, são associadas a métodos adaptáveis, contrariamente às metodologias tradicionais, normalmente, associadas a forte disciplina e planeamento, ver Figura 6 e devem atender à natureza dinâmica do processo de concepção de um sistema de software.

Assim, as metodologias ágeis deverão estar habilitadas para lidarem com todo o tipo de mudanças que possam surgir ao longo do ciclo de vida do sistema, designadamente alterações aos requisitos, inicialmente identificados (Semedo, 2012).

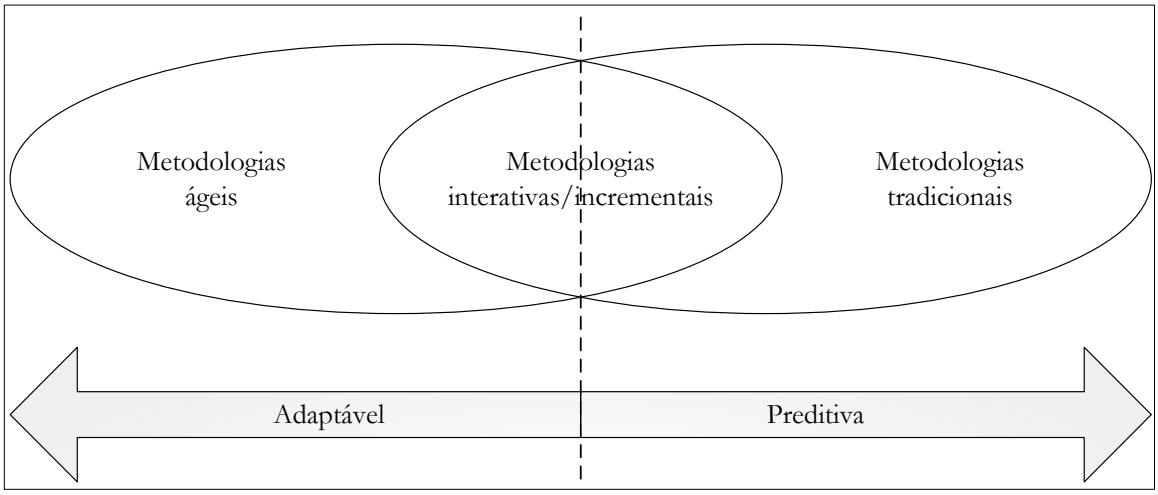


Figura 6 – Plasticidade associada às metodologias ágeis/tradicionais (Barbosa et al., 2007)

Há várias abordagens que podem ser utilizadas no âmbito das metodologias ágeis (p. ex. *Extreme Programming* (XP), *Adaptative Software Development* (ASD), *Feature Driven Development* (FDD), LEAN, *Dynamic Systems Development Method* (DSDM), FAMILY CRYSTAL e SCRUM). (Ferreira, 2013). Todas estas abordagens podem ser consideradas no âmbito do que é designado por desenvolvimento interativo e incremental (Larman & Basili, 2003) e baseia-se na utilização de um conjunto de atividades adequadas para projetos de curto prazo e para requisitos que podem mudar frequentemente (Schwaber & Sutherland, 2013).

[18] http://agilemanifesto.org/

Assim e atendendo às características e ao contexto em que é feito o desenvolvimento e a implementação do protótipo associado ao sistema CYCLOPS vamos utilizar, genericamente, uma abordagem de desenvolvimento interativo e incremental que segue um processo iterativo de desenvolvimento baseado em estágios de desenvolvimento e em que se vai confrontando a evolução do projeto com as especificações apresentadas pelo cliente, em função das quais vão sendo definidas novas funcionalidades. A utilização desta abordagem, enquadrada nas metodologias ágeis, será acompanhada por uma adequada gestão do risco que lhe está sempre associado.

3.4 Ciclo de vida

Considerando que este projeto de estágio e o respetivo desenvolvimento e implementação do protótipo que lhe está associado se enquadra na área da engenharia de software e que a sua natureza tem uma óbvia componente de investigação/experimentação, não se dispõe à partida do conhecimento necessário que permita dar adequada resposta ao problema identificado e às respetivas questões de investigação. Assim, conforme já referido anteriormente, optámos pela utilização de metodologias ágeis no desenvolvimento do protótipo do sistema CYCLOPS.

No âmbito das metodologias ágeis, atendendo às características do projeto, conforme referido na secção anterior, será utilizado o modelo de desenvolvimento interativo e incremental, ver Figura 7. Este modelo pressupõe que percebido o contexto do problema se identifiquem os requisitos não funcionais e funcionais, assim como os atributos de qualidade e a respetiva arquitetura. Percebido o problema, identificados os requisitos e encontrada a arquitetura adequada, serão realizadas reuniões periódicas onde será avaliado o trabalho executado no período correspondente e feitas as correções necessárias.

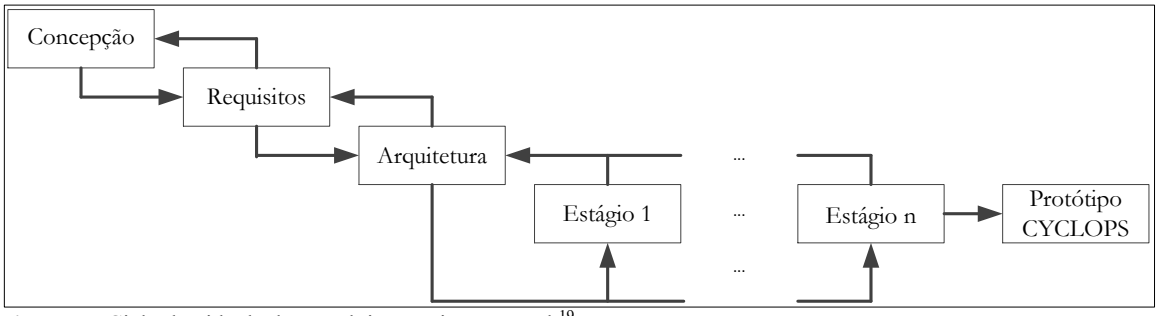


Figura 7 - Ciclo de vida da desenvolvimento incremental [19]

3.5 Planeamento

Conforme referido, na secção anterior, o presente projeto foi desenvolvido em duas fases: a primeira fase que correspondeu ao 1º semestre do ano letivo 2015/2016 e decorreu entre setembro de 2015 e fevereiro de 2016 e a segunda fase, correspondente ao 2º semestre do mesmo ano letivo, que decorreu entre fevereiro e junho de 2016. Para cada uma destas fases foi feito o respetivo planeamento apresentando-se na Figura 8 o diagrama de Gantt relativo à primeira fase e na Figura 9 o diagrama de Gantt relativo à segunda fase. Conforme já referido anteriormente, a primeira fase foi dedicada essencialmente a perceber os objetivos deste projeto e a preparar a base necessária para o trabalho de desenvolvimento e implementação que teve lugar na segunda fase.

[19] https://profchs.files.wordpress.com/2012/05/ciclos.pdf

3.5.1 Primeira fase

Durante a primeira fase foram cumpridas diversas atividades, em consonância com as orientações da equipa de orientação. Assim, nesta fase, foram agendadas e realizadas reuniões periódicas (com periodicidade aproximada de 3 semanas) entre o estagiário e o colaborador da CRITICAL Software, SA que desempenhou o papel de cliente. Nestas reuniões, o estagiário apresentou o trabalho realizado em cada um dos períodos referidos, e obteve o respetivo *feedback* por parte do cliente permitindo ir avaliando se o trabalho desenvolvido se enquadrava com a expectativa do cliente. Portanto, cada uma das etapas do projeto foi discutida e validada com o cliente, avançando-se para a etapa seguinte do projeto somente após validação, por parte do mesmo, da etapa anterior.

Conforme, já referido anteriormente, na Figura 8 é apresentado o diagrama de Gantt relativo ao trabalho desenvolvido durante a primeira fase, nele poderemos ver que grande parte do tempo, afeto a esta fase, foi absorvida pela revisão da literatura e análise de trabalho relacionado com o âmbito da pesquisa de informação empresarial armazenada em repositórios corporativos. Ainda no decorrer desta fase, com o conhecimento adquirido, iniciámos a preparação de levantamento e especificação de requisitos recorrendo a uma abordagem etnográfica e à elaboração de um modelo de questionário que, complementando a abordagem etnográfica, permitisse sentir a sensibilidade dos colaboradores da iTGROW e da CRITICAL Software, SA relativamente à importância da existência de um sistema empresarial de pesquisa.

Tendo em atenção os trabalhos que identificam e caracterizam o enquadramento teórico do tema deste projeto e ainda a análise de diversos trabalhos relacionados com o tema proposto no âmbito deste projeto, foi reunido conhecimento necessário que permitiu clarificar o desafio, associado a este projeto, detalhar os seus objetivos e identificar algumas questões que, após respondidas, ajudarão a encontrar a melhor solução para responder ao problema, cumprindo assim os objetivos identificados. Clarificado o problema, detalhados os objetivos e identificadas algumas questões consideradas relevantes, foram reunidas, igualmente, as condições necessárias para dar início ao processo de identificação e especificação dos requisitos necessários para desenvolver e implementar o protótipo do sistema CYCLOPS.

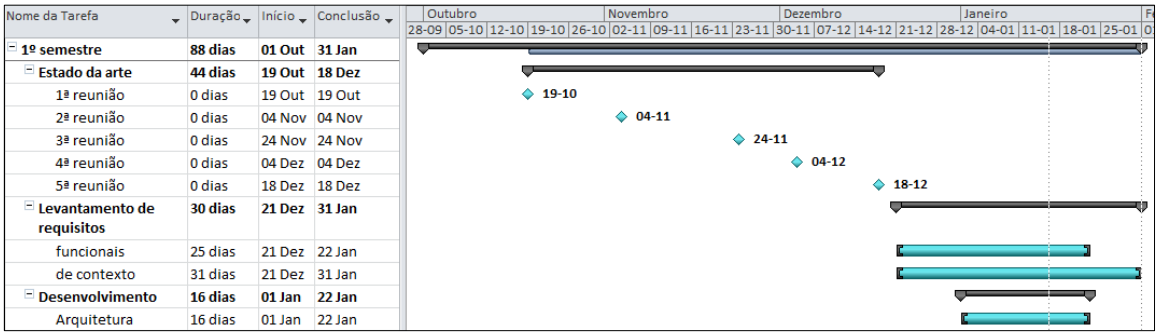


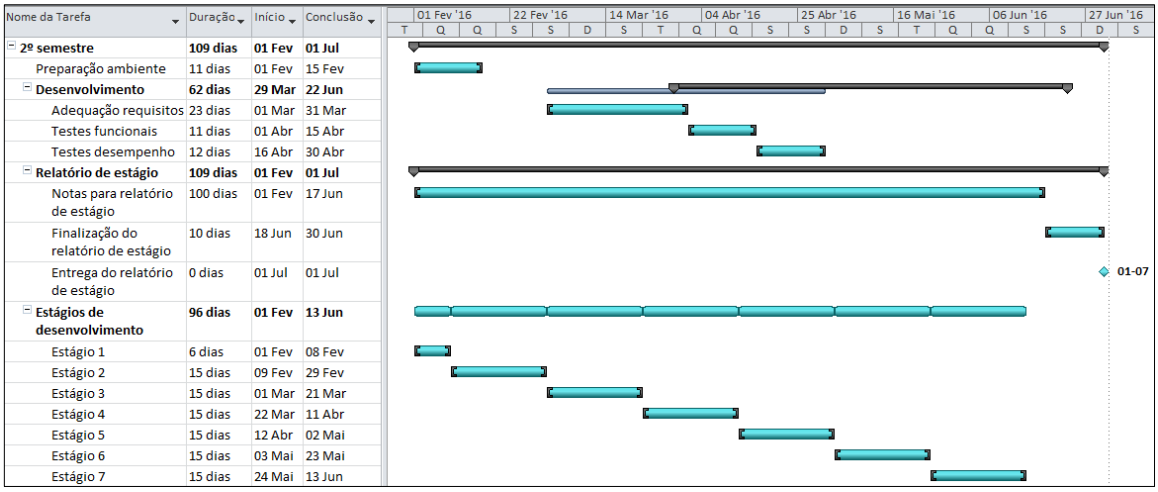
Figura 8 - Diagrama de Gantt relativo às atividades realizadas durante o primeiro semestre

3.5.2 Segunda fase

O diagrama de Gantt apresentado na Figura 9 mostra a evolução das diversas atividades que foi necessário cumprir no decorrer da segunda fase deste projeto, que consistitu essencialmente na execução de atividades relacionadas com o desenvolvimento e implementação do protótipo do sistema CYCLOPS.

Tendo em atenção o diagrama de Gantt, mostrado na Figura 9, podemos verificar que as principais atividades desenvolvidas neste período foram as seguintes: preparação do

ambiente de desenvolvimento; desenvolvimento e implementação do protótipo (esta fase inclui a adequação de requisitos e a realização de testes funcionais e de desempenho); relatório de estágio e estágios de desenvolvimento.



Nome da Tarefa	Duração	Início	Conclusão
2º semestre	**109 dias**	**01 Fev**	**01 Jul**
Preparação ambiente	11 dias	01 Fev	15 Fev
Desenvolvimento	**62 dias**	**29 Mar**	**22 Jun**
Adequação requisitos	23 dias	01 Mar	31 Mar
Testes funcionais	11 dias	01 Abr	15 Abr
Testes desempenho	12 dias	16 Abr	30 Abr
Relatório de estágio	**109 dias**	**01 Fev**	**01 Jul**
Notas para relatório de estágio	100 dias	01 Fev	17 Jun
Finalização do relatório de estágio	10 dias	18 Jun	30 Jun
Entrega do relatório de estágio	0 dias	01 Jul	01 Jul
Estágios de desenvolvimento	**96 dias**	**01 Fev**	**13 Jun**
Estágio 1	6 dias	01 Fev	08 Fev
Estágio 2	15 dias	09 Fev	29 Fev
Estágio 3	15 dias	01 Mar	21 Mar
Estágio 4	15 dias	22 Mar	11 Abr
Estágio 5	15 dias	12 Abr	02 Mai
Estágio 6	15 dias	03 Mai	23 Mai
Estágio 7	15 dias	24 Mai	13 Jun

Figura 9 - Diagrama de Gantt representativo das atividades a desenvolver durante o segundo semestre

Para terem lugar durante a segunda fase deste projeto, que ocorreu entre fevereiro e junho de 2016, conforme se pode verificar através do diagrama de Gantt, apresentado na Figura 9, foi prevista a realização de diversos estágios de desenvolvimento.

Em cada um destes estágios de desenvolvimento, deverão ter lugar um conjunto de atividades específicas, cuja realização em conformidade, com o planeado, permitirá cumprir os objetivos inicialmente delineados. Pode assim, verificar-se a importância dos estágios de desenvolvimento, sendo assim de toda a importância apresentar a descrição genérica das tarefas que serão executadas em cada um deles.

Assim, de seguida é apresentada a descrição genérica do trabalho a realizar em cada um dos estágios de desenvolvimento identificados no diagrama de Gantt apresentado na Figura 9. Na secção 3.4 onde é comparado o planeado com o trabalho realizado, nesta segunda fase, é feita referência ao cumprimento deste plano de trabalhos.

(01/fevereiro/2016 a 08/fevereiro/2016) → Estágio 1

- neste período foi feita a planificação dos trabalhos relativos ao desenvolvimento do protótipo do sistema CYCLOPS.

(09/fevereiro/2016/ a 29/fevereiro/2016) → Estágio 2

- desenvolvimento do *crawler*, para recolher informação no repositório *SVN*;
- avaliação da plataforma mais adequada para gestão do transporte e distribuição de documentos entre os diversos blocos que constituem a arquitetura do protótipo.

(30/fevereiro/2016/ a 21/março/2016) → Estágio 3

- implementação das filas de espera, utilizando a plataforma Apache Kafka;
- desenvolvimento do *parser*;
- integração numa perspetiva sistémica.

(22/março/2016/ a 11/abril/2016) → Estágio 4

- implementação das filas de espera, entre o *parser* e o Apache Solr;
- desenvolvimento do *indexer* e *parser* assim como do respetivo índex, utilizando a plataforma de pesquisa empresarial Apache Solr;

- desenvolvimento de mecanismo de ACL;
- desenvolvimento do *crawler* para o repositório *AsknowledgeDb*;

(12/abril/2016/ a 02/maio/2016) → Estágio 5

- avaliação e melhoria do código desenvolvido nas fases anteriores;
- desenvolvimento de interface para o utilizador.

(03/maio/2016/ a 23/maio/2016) → Estágio 6

- avaliação do protótipo desenvolvido;
- melhoria da interface para o utilizador.

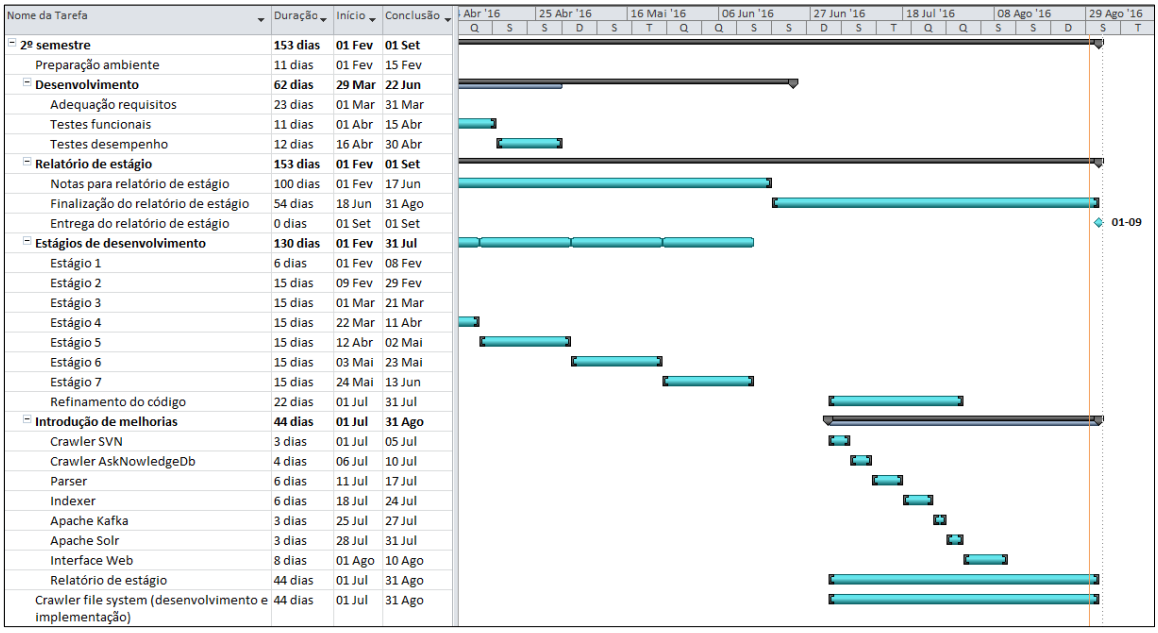
(24/maio/2016/ a 13/junho/2016) → Estágio 7

- trabalhar aspetos que possam melhorar o protótipo desenvolvido.

3.6 Planeamento versus execução

Durante o processo de desenvolvimento e implementação do protótipo CYCLOPS houve o cuidado de, para além de identificar e avaliar o risco associado a cada um dos estágios, proceder à verificação da execução em função do planeado.

Este procedimento não permitiu identificar grandes alterações ao plano inicial com exceção de na parte final do projeto se ter entendido ser vantajoso, por trazer melhorias ao trabalho desenvolvido, prolongar os trabalhos pelos meses de julho e agosto. Este adiamento na conclusão do projeto permitiu, conforme já referimos introduzir algumas melhorias, designadamente no refinamento do código desenvolvido ao nível dos *crawlers*, do *parser* e do *indexer*, assim como ao nível da interface *Web*.



Nome da Tarefa	Duração	Início	Conclusão
2º semestre	**153 dias**	**01 Fev**	**01 Set**
Preparação ambiente	11 dias	01 Fev	15 Fev
Desenvolvimento	**62 dias**	**29 Mar**	**22 Jun**
Adequação requisitos	23 dias	01 Mar	31 Mar
Testes funcionais	11 dias	01 Abr	15 Abr
Testes desempenho	12 dias	16 Abr	30 Abr
Relatório de estágio	**153 dias**	**01 Fev**	**01 Set**
Notas para relatório de estágio	100 dias	01 Fev	17 Jun
Finalização do relatório de estágio	54 dias	18 Jun	31 Ago
Entrega do relatório de estágio	0 dias	01 Set	01 Set
Estágios de desenvolvimento	**130 dias**	**01 Fev**	**31 Jul**
Estágio 1	6 dias	01 Fev	08 Fev
Estágio 2	15 dias	09 Fev	29 Fev
Estágio 3	15 dias	01 Mar	21 Mar
Estágio 4	15 dias	22 Mar	11 Abr
Estágio 5	15 dias	12 Abr	02 Mai
Estágio 6	15 dias	03 Mai	23 Mai
Estágio 7	15 dias	24 Mai	13 Jun
Refinamento do código	22 dias	01 Jul	31 Jul
Introdução de melhorias	**44 dias**	**01 Jul**	**31 Ago**
Crawler SVN	3 dias	01 Jul	05 Jul
Crawler AskNowledgeDb	4 dias	06 Jul	10 Jul
Parser	6 dias	11 Jul	17 Jul
Indexer	6 dias	18 Jul	24 Jul
Apache Kafka	3 dias	25 Jul	27 Jul
Apache Solr	3 dias	28 Jul	31 Jul
Interface Web	8 dias	01 Ago	10 Ago
Relatório de estágio	44 dias	01 Jul	31 Ago
Crawler file system (desenvolvimento e implementação)	44 dias	01 Jul	31 Ago

Figura 10 - Diagrama de *Gantt* relativo à execução do trabalho desenvolvido na segunda fase do projeto

Conforme podemos verificar, comparando os diagramas de Gantt mostrados na Figura 12 e na Figura 10 (trabalho planeado e trabalho realizado, respetivamente) relativos à segunda fase do projeto, coincidente com o 2º semestre do ano letivo 2015/2016, no essencial foi cumprido o que estava planeado com exceção da parte final em que foi decidido prolongar os trabalhos durante os meses de julho e agosto para poder introduzir algumas melhorias ao

protótipo desenvolvido e desenvolver e implementar um novo *crawler* para procura e recolha de informação armazenada num repositório *file system*. Assim, as alterações ao planeamento inicial somente são visíveis no relativamente à alteração da data de conclusão do protótipo, passou para 1 de setembro, à introdução de melhorias e ao desenvolvimento de um *crawler* para repositórios *file system* a decorrerem nos meses de julho e agosto.

Foi ainda entendido oportuno, neste período, desenvolver um novo *crawler* (não previsto no planeamento inicial) para procura e recolha de informação num repositório *file system*. O diagrama de Gantt que inclui as alterações ao planeamento inicial é mostrado na Figura 10.

3.7 Gestão do risco

Em qualquer projeto, deve ser dada particular atenção à gestão do risco que lhe está sempre associado. Esta gestão deve ser feita através de uma abordagem sistemática e contínua com o objetivo da sua minimização. O risco pode ser visto na perspetiva de um par causa-efeito, a causa é uma ameaça e o efeito é a consequência da concretização dessa ameaça, conforme mostrado na Figura 11 e que mostra os seus componentes (Alberts & Dorofee, 2010).

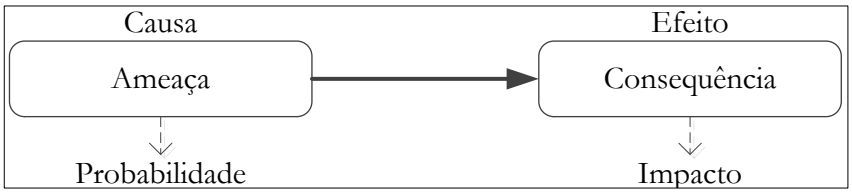


Figura 11 - Componentes do risco (Alberts & Dorofee, 2010)

O ciclo de vida de um processo de gestão de risco pode ser visto como representado na Figura 12, adaptada de (Alberts & Dorofee, 2010).

Figura 12 - Etapas do processo de gestão do risco (Alberts & Dorofee, 2010)

Conforme podemos verificar, analisando a Figura 12, a primeira fase do ciclo de vida de um processo de gestão do risco consiste na avaliação do risco, seguindo-se o planeamento das ações que tenham por objetivo a sua minimização e, finalmente, a minimização da exposição ao risco através da implementação de adequados mecanismos de controlo.

Assim, conforme referido no parágrafo anterior, podemos associar a gestão do risco de um determinado projeto a três processos, designadamente: (1) identificação e análise na fase de avaliação; (2) planeamento na fase de planeamento e (3) monitorização e controlo, na fase de mitigação. Estes processos devem ser atualizados durante o ciclo de vida do projeto, com o objetivo de aumentar a probabilidade e o impacto dos eventos positivos (favoráveis ao projeto) e diminuir a probabilidade e o impacto dos eventos negativos (adversos ao projeto) (Project Management Institute, 2008). Este conceito de gestão de risco, em que são trabalhadas as vertentes positiva e negativa do risco, é apresentado pela *British Standard* BS 6079-1:2000, que define risco como uma combinação da probabilidade ou frequência da ocorrência de uma ameaça ou oportunidade e o impacto das consequências desta ocorrência (Bonanomi, Silva, & Rocha, 2012).

Neste projeto, a atenção é focada nos eventos negativos (adversos ao projeto), considerando que, normalmente, a implementação de um processo de gestão do risco se baseia na utilização de uma matriz de risco, sendo, na opinião de muitos, o caso mais representativo desta utilização a análise modal de falhas, abordagem muito utilizada na gestão do risco.

3.7.1 Matriz de risco

A exposição ao risco em função da probabilidade e do impacto pode ser assim calculada (Bonanomi et al., 2012):

$$E = P \times I$$

em que:
E – exposição;
P – probabilidade;
I – impacto.

São definidas duas escalas, uma para a probabilidade de ocorrência e outra para o impacto sobre o andamento do projeto. Para este caso, em concreto, é utilizada a seguinte escala para a probabilidade de ocorrência (0,1; 0,3; 0,5; 0,7 e 0,9) em que 0,1 corresponde a "muito improvável" e 0,9 corresponde a "quase certeza", ver Tabela 1. Esta escala, constituída por valores que pertencem ao intervalo [0...1], é sugerida por Bonanomi *et al* (2012).

PROBABILIDADE	
Muito improvável	0,1
Improvável	0,3
Provável	0,5
Muito provável	0,7
Quase certeza	0,9

Tabela 1 - Escala relativa á probabilidade de ocorrência (Bonanomi et al., 2012)

De forma similar à probabilidade de ocorrência, para o impacto sobre o andamento do projeto é utilizada a seguinte escala (0,05; 0,1; 0,2; 0,4; 0,8) em que 0,05 corresponde "muito baixo" e 0,8 corresponde a "muito alto", ver Tabela 2. Esta escala é constituída, igualmente, por valores que pertencem ao intervalo [0...1] e é sugerida por Bonanomi *et al* (2012).

IMPACTO	
Muito baixo	0,05
Baixo	0,1
Significativo	0,2
Alto	0,4
Muito alto	0,8

Tabela 2 - Escala relativa ao impacto que pode existir (Bonanomi et al., 2012)

Estas duas escalas devem ser definidas pela equipa responsável pela gestão do risco e podem ser lineares ou não lineares, dependendo de vários fatores (Bonanomi et al., 2012; Project Management Institute, 2008). O produto da pontuação atribuída aos riscos identificados, relativamente ao seu impacto e probabilidade de ocorrência, permite obter uma matriz de risco devidamente preenchida, ver Tabela 3, destacando a zona crítica (zona vermelha), a zona neutra (zona amarela) e a zona aceitável (zona verde). Se o valor da exposição ao risco cair na zona crítica devemos implementar mecanismos que minimizem essa exposição, se cair na zona amarela poderemos investir em mecanismos que reduzam essa exposição se a análise custo x benefício for favorável a essa opção, finalmente se o valor da exposição

estiver na zona verde, o risco é aceitável e não será necessário investir na implementação de mecanismos que o reduzam.

Probabilidade						
	0,9	0,05	0,09	0,18	0,36	0,72
	0,7	0,04	0,07	0,14	0,28	0,56
	0,5	0,03	0,05	0,10	0,20	0,40
	0,3	0,02	0,03	0,06	0,12	0,24
	0,1	0,01	0,01	0,02	0,04	0,08
		0,05	**0,1**	**0,2**	**0,4**	**0,8**
		Impacto				

Tabela 3 - Matriz de risco (probabilidade x impacto) (Bonanomi et al., 2012)

3.7.2 Identificação e avaliação do risco

no âmbito de cada um dos estágios de desenvolvimento referidos na secção anterior procedemos à identificação dos riscos que lhe estão associados e à avaliação da respetiva exposição ao risco, em função da probabilidade de ocorrência e do impacto que os referidos riscos poderão ter no desenvolvimento do projeto.

Estágio 1

O período relativo a este estágio, em que foi feita a planificação dos trabalhos a desenvolver na segunda fase deste projeto, coincidiu com a apresentação do relatório intermédio. Na altura, foram identificadas várias fragilidades (riscos) associadas ao trabalho desenvolvido até então. Do levantamento feito foram identificadas quatro fragilidades (riscos) relevantes: (1) identificação dos requisitos por consolidar; (2) indefinição da arquitetura a utilizar; (3) falhas no levantamento do estado da arte e (4) indefinição quanto às tecnologias a utilizar.

Considerando estes riscos e as escalas associadas à probabilidade da sua ocorrência e ao seu possível impacto no desenvolvimento do projeto, foram identificados e priorizados os riscos apresentados na Tabela 4. Para cada um dos riscos identificados é proposto um mecanismo de controlo cujo objetivo é minimizar a sua exposição.

IDENTIFICAÇÃO E PRIORIZAÇÃO DO RISCO							
ID	**Ameaça**	**Consequência**	**P**	**I**	**E**	**Priorização**	**Controlo recomendado**
2	Atraso na identificação e definição da arquitetura a utilizar	Atraso no desenvolvimento do projeto	0,7	0,4	0,28	1	Consolidar o levantamento de requisitos não funcionais e funcionais e identificar os blocos fundamentais da arquitetura.
1	Atraso na identificação dos requisitos não funcionais e funcionais	Atraso no desenvolvimento do projeto	0,5	0,4	0,20	2	Complementar as respostas ao inquérito já realizado com reuniões com a equipa de I&D.
3	Atraso no levantamento do estado da arte	Atraso no desenvolvimento do projeto	0,5	0,2	0,10	3	Aprofundar a revisão bibliográfica, não só ao nível de conceitos, mas também ao nível de trabalho relacionado.
4	Atraso na identifcação das tecnologias a utilizar	Atraso no desenvolvimento do projeto	0,5	0,2	0,10	3	Paralelamente à consolidação dos blocos que constituem a arquitetura, definir as tecnologias a utilizar.

Tabela 4 - Identificação e priorização de riscos que podem ocorrer no decurso do *estágio 1*

Analisando a tabela anterior, podemos verificar que um dos riscos identificados tem uma exposição de 0,28, que de acordo com a matriz de risco apresentada na Tabela 3 está na zona vermelha (zona crítica). Assim, é necessário dar particular atenção à minimização da exposição associada a este risco investindo nos adequados mecanismos de controlo. Dos

restantes três riscos identificados um tem uma exposição de 0,20 e dois têm uma exposição de 0,10 que, de acordo com a matriz de risco, se situam na zona amarela (zona neutra), neste caso deve ser feita uma análise custo x benefício para decidir se vale a pena investir nos mecanismos de controlo adequados e que estão referidos na mesma matriz, neste caso particular, considerando que a relação custo x benefício é favorável ao cliente, vale a pena investir na minimização do risco.

Estágio 2

Durante o período relativo ao estágio 2 foram implementados os mecanismos de controlo recomendados para minimizar a exposição ao risco avaliada no estágio 1. Os mecanismos implementados reduziram significativamente esta exposição uma vez que ficou definida a arquitetura a utilizar no desenvolvimento do protótipo, foi consolidada a lista de requisitos não funcionais e funcionais, foi feito o levantamento de trabalho relacionado e foram identificadas as tecnologias a utilizar. No entanto surgiram outros riscos cuja probabilidade de ocorrência e impacto no desenvolvimento do projeto conduziram a novas exposições críticas que carecem de ser minimizadas através da implementação de adequados mecanismos de controlo.

Nesta fase foi desenvolvido o *crawler* para procura e recolha de informação no repositório *SVN* e foram comparadas algumas das possíveis plataformas para gestão e transporte dos documentos recolhidos por este *crawler* e também pelo *crawler* para recolha de informação no repositório *AsknowledgeDB*, a desenvolver no período relativo ao estágio 4. Assim, nesta fase, o risco que consideramos mais relevante está relacionado com o eventual incumprimento do prazo que atribuímos ao desenvolvimento e implementação do *crawler* para o repositório *SVN*.

Considerando este risco e as escalas associadas à probabilidade da sua ocorrência e do seu impacto no desenvolvimento do projeto, foi identificado e priorizado o risco, conforme apresentada na Tabela 5 onde é visível a criticidade da exposição e o controlo recomendado para que a mesma possa ser minimizada.

IDENTIFICAÇÃO E PRIORIZAÇÃO DO RISCO							
ID	**Ameaça**	**Consequência**	**P**	**I**	**E**	**Priorização**	**Controlo recomendado**
1	Não cumprimento do prazo estabelecido para desenvolvimento e implementação do crawlers para o repositório SVN	Atraso no desenvolvimento do projeto	0,7	0,4	0,28	1	Apresentação semanal do trabalho realizado e verificar cumprimento do planeado.

Tabela 5 - Identificação e priorização do risco que pode ocorrer no período relativo ao estágio 2

Analisando a tabela anterior, pode ver-se que o único risco identificado tem uma exposição de 0,28, que de acordo com a matriz de risco apresentada na Tabela 3 está na zona vermelha (zona crítica). Assim, é necessário dar particular atenção à minimização da exposição a este risco investindo nos adequados mecanismos de controlo.

Estágio 3

No período relativo ao estágio 3 foi dado especial cuidado à implementação de filas, mecanismo para gestão de mensagens, utilizando o Apache Kafka. Em simultâneo foi iniciado, igualmente, o desenvolvimento do *parser*, nunca descurando a melhoria do *crawler* desenvolvido no estágio 2 e a necessidade da sua interação com outros módulos do protótipo em desenvolvimento.

À semelhança do estágio 2, também aqui, deve ser destacado, o risco relacionado com o eventual incumprimento dos prazos definidos para implementação das filas e com o desenvolvimento e implementação do *parser* não esquecendo que poderão, eventualmente,

surgir ainda problemas com o *crawler*, devido à necessidade da sua integração com outros componentes do sistema. Assim foram identificados e priorizamos os riscos apresentados na Tabela 6.

IDENTIFICAÇÃO E PRIORIZAÇÃO DO RISCO							
ID	**Ameaça**	**Consequência**	**P**	**I**	**E**	**Priorização**	**Controlo recomendado**
2	Não cumprimento do prazo estabelecido para desenvolvimento e implementação do parser.	Atraso no desenvolvimento do projeto	0,7	0,4	0,28	1	Apresentação semanal do trabalho realizado e verificar cumprimento do planeado.
3	Não cumprimento do prazo estabelecido para desenvolvimento e implementação das filas.	Atraso no desenvolvimento do projeto	0,7	0,4	0,28	1	Apresentação semanal do trabalho realizado e verificar cumprimento do planeado.
1	Falhas na integração do crawler do repositório SVN com restantes módulos.	Atraso no desenvolvimento do projeto	0,3	0,2	0,06	2	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.

Tabela 6 - Identificação e priorização de riscos que podem ocorrer no período relativo ao estágio 3

Analisando a Tabela 6, pode verificar-se que dois dos riscos identificados têm uma exposição de 0,28, que de acordo com a matriz de risco apresentada na Tabela 3 está na zona vermelha (zona crítica). Assim, é necessário dar particular atenção à minimização deste risco investindo nos adequados mecanismos de controlo. O terceiro risco, apresentado na Tabela 6, apresenta um nível de exposição de 0,06, situando-se na zona amarela (zona neutra), neste caso deve ser feita uma análise custo x benefício para decidir se vale a pena investir no desenvolvimento e implementação de mecanismos de controlo adequados e que estão referidos na mesma matriz, neste caso particular, considerando que a relação custo x benefício é favorável ao cliente, vale a pena investir na minimização do risco.

Estágio 4

No período de tempo associado a este estágio, para além de ser desenvolvido o *crawler* para o repositório *AsknowledgeDb*, cuidámos da implementação das filas entre o *parser* e o Apache Solr, do desenvolvimento do *indexer* e também do desenvolvimento do mecanismo de ACL. Também foram tidas em atenção eventuais falhas na integração dos módulos desenvolvidos nas fases anteriores com restantes módulos que constituem o protótipo que se encontra em desenvolvimento.

De forma análoga ao que foi referido, relativamente ao estágio 2 e ao estágio 3, os riscos mais relevantes, relativamente à sua probabilidade de ocorrência e ao seu impacto sobre o desenvolvimento do projeto, continuam a ser o eventual não cumprimento de prazos de desenvolvimento e problemas que possam surgir na fase de integração dos diversos módulos. Assim, foram identificados e priorizados os riscos apresentados na Tabela 7.

IDENTIFICAÇÃO E PRIORIZAÇÃO DO RISCO							
ID	**Ameaça**	**Consequência**	**P**	**I**	**E**	**Priorização**	**Controlo recomendado**
5	Não cumprimento do prazo estabelecido para desenvolvimento e implementação dos mecanismos ACL.	Atraso no desenvolvimento do projeto	0,7	0,8	0,56	1	Apresentação semanal do trabalho realizado e verificar cumprimento do planeado.
4	Não cumprimento do prazo estabelecido para desenvolvimento e implementação do indexer.	Atraso no desenvolvimento do projeto	0,7	0,4	0,28	2	Apresentação semanal do trabalho realizado e verificar cumprimento do planeado.
6	Não cumprimento do prazo estabelecido para desenvolvimento e implementação do crawler (repositório AsknowledgeDb)	Atraso no desenvolvimento do projeto	0,7	0,4	0,28	2	Apresentação semanal do trabalho realizado e verficar cumprimento do planeado.
1	Falhas na integração do parser com restantes módulos.	Atraso no desenvolvimento do projeto	0,3	0,2	0,06	3	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.
2	Falhas na integração dos filas com restantes módulos.	Atraso no desenvolvimento do projeto	0,3	0,2	0,06	3	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.
3	Falhas na integração do crawler (repositório SVN) com restantes módulos.	Atraso no desenvolvimento do projeto	0,3	0,1	0,03	4	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.

Tabela 7 - Identificação e priorização de riscos que podem ocorrer no período relativo ao estágio 4

Analisando a Tabela 7, pode ver-se que um dos riscos com prioridade 1 tem um nível de exposição de 0,56 e dois apresentam prioridade 2 com um nível de exposição correspondente a 0,28. Assim, de acordo com a matriz de risco apresentada na Tabela 3 estes riscos encontram-se na zona vermelha (zona crítica), sendo necessário dar particular atenção à sua minimização investindo nos adequados mecanismos de controlo. Existem ainda dois riscos com prioridade 3 cujo nível de exposição é de 0,06 que de acordo com a matriz de risco referida se situam na zona amarela (zona neutra), neste caso deve ser feita uma análise custo x benefício para decidir se vale a pena investir nos mecanismos de controlo adequados e que estão referidos na mesma matriz, neste caso particular, considerando que os custos a afetar ao cliente são baixos, vale a pena investir na minimização do risco. Finalmente há ainda a considerar um riscos com prioridade 4 cujo nível de exposição é de 0,03 situando-se assim na zona verde (zona de risco aceitável), neste caso não é necessário preocupação especial com a sua minimização.

Estágio 5

No período relativo ao estágio 5 foram trabalhados os aspetos relativos à melhoria do código associado aos módulos desenvolvidos nas fases anteriores e ainda ao desenvolvimento da interface *Web*.

Nesta fase, para além dos riscos que poderão ocorrer com a integração dos módulos já desenvolvidos, o risco identificado como mais relevante está relacionado com o eventual incumprimento do prazo relativo ao desenvolvimento da interface *Web*, conforme apresentado na Tabela 8.

IDENTIFICAÇÃO E PRIORIZAÇÃO DO RISCO							
ID	**Ameaça**	**Consequência**	**P**	**I**	**E**	**Priorização**	**Controlo recomendado**
5	Falhas na integração dos mecanismos ACL, com restantes módulo.	Falhas na segurança	0,7	0,4	0,28	1	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.
4	Falhas na integração do indexer com restantes módulos	Atraso no desenvolvimento do projeto	0,3	0,2	0,06	2	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.
1	Falhas na integração do parser com restantes módulos.	Atraso no desenvolvimento do projeto	0,3	0,1	0,03	3	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.
2	Falhas na integração dos filas com restantes módulos.	Atraso no desenvolvimento do projeto	0,3	0,1	0,03	3	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.
3	Falhas na integração dos crawlers com restantes módulos.	Atraso no desenvolvimento do projeto	0,1	0,1	0,01	4	Apresentação semanal do trabalho realizado (numa perspetiva de sistema) e verificar cumprimento do planeado.

Tabela 8 - Identificação e priorização de riscos que podem ocorrer no decurso do estágio 5

Analisando a tabela anterior pode verificar-se que relativamente ao risco com prioridade 1 corresponde um nível de exposição de 0,28, que de acordo com a matriz de risco apresentada na Tabela 3 se situa na zona vermelha (zona crítica). Assim, é necessário dar particular atenção à minimização deste risco investindo nos adequados mecanismos de controlo. Ao risco com prioridade 2 corresponde um nível de exposição de 0,06 que de acordo com a matriz de risco se situa na zona amarela (zona neutra), neste caso deve ser feita uma análise custo x benefício para decidir se vale a pena investir nos mecanismos de controlo adequados e que estão referidos na mesma matriz, neste caso particular, considerando que a relação custo x benefício é favorável ao cliente, vale a pena investir na sua minimização. Finalmente podemos verificar a existência de dois riscos com prioridade 3 a que corresponde um nível de exposição de 0,03 e de um risco com prioridade 4 a que corresponde um nível de exposição de 0,01. Todos estes riscos se situam na zona verde (zona de risco aceitável), neste caso não é necessário preocupação especial com a sua minimização.

Estágio 6 e estágio 7

Finalmente no estágio 6 e no estágio 7 não há riscos relevantes a considerar devendo, contudo, haver o cuidado de manter os níveis de exposição tão baixas quanto possível, manutenção na zona verde (zona de risco aceitável), utilizando e mantendo ativos os mecanismos de controlo sugeridos.

No período relativo a estes dois estágios de desenvolvimento não está prevista a construção de novos módulos, sendo essencialmente destinado à realização de testes e verificação do protótipo desenvolvido e à melhoria do código relativo aos módulos desenvolvidos e implementados nas fases anteriores.

[Esta página foi intencionalmente deixada em branco]

Capítulo 4
Requisitos e Arquitetura

Conforme já referido no Capítulo 3, no desenvolvimento deste projeto, foi esencialmente utilizada uma abordagem etnográfica para levantamento de requisitos e desenvolvimento da arquitetura. Durante todo este processo, à semelhança do que ocorreu durante todo o período em que decorreu a realização deste estágio, foram realizadas várias reuniões de trabalho com a participação efetiva do orientador por parte da CRITICAL Software, SA, assim como de outros elementos que fazem parte da equipa de I&D que acolheu o estagiário.

Uma arquitetura para implementação de um protótipo do sistema CYCLOPS deve descrever os principais componentes do sistema: as suas ligações e as suas interações assim como a sua natureza. Assim, a sua avaliação deve ser feita de forma interativa desde a fase inicial até à sua implementação prática, trazendo esta prática grande valor ao sistema (Barbacci, Klein, longstaff, & Weinstock, 1995).

A arquitetura para o nosso protótipo será definida a partir da identificação e especificação dos requisitos não funcionais e funcionais e dos respetivos atributos de qualidade não esquecendo o contributo de um alargado conjunto de *building blocks* para implementação deste tipo de sistemas assim como da análise de várias elementos arquiteturais relativos aos diversos módulos que constituem qualquer sistema de procura de informação empresarial.

4.1 Especificação de requisitos

Um dos grandes problemas que se colocam no desenvolvimento de qualquer sistema de software está relacionado com a identificação e definição dos seus requisitos, que determinam de que forma o sistema deverá responder (Sommerville, 2007).

Estes requisitos quando são influenciados pelo que devem fornecer, pela forma como devem reagir e pela forma como se devem comportar, são designados de requisitos funcionais. Outros requisitos determinados por restrições sobre os serviços ou funções que devem ser disponibilizadas pelo sistema, são designados de requisitos não funcionais (Sommerville, 2007). Assim, neste relatório, a atenção será focada na identificação e definição de requisitos não funcionais e funcionais, sem prejuízo de fazer essa análise incidir sobre cada um dos componentes que constitui o protótipo a desenvolver e implementar no âmbito deste projeto.

Neste processo para especificação dos requisitos, que devem ser observados pelo protótipo associado ao sistema CYCLOPS, serão seguidas de perto as recomendações da IEEE Computer Society (1998). Este protótipo tem por objetivo a pesquisa de informação empresarial em dois repositórios já referidos anteriormente: (1) o repositório *SVN*, e (2) o repositório *AsknowledgeDb* ajudando, assim, os colaboradores da CRITICAL Software, SA a

rapidamente encontrarem a informação de que necessitam para desempenharem as suas atividades profissionais.

Para que os objetivos associados ao desenvolvimento deste protótipo sejam atingidos devem ser garantidos determinados pressupostos que terão expressão na especificação dos requisitos apresentados neste capítulo, que conforme já referimos anteriormente vamos dividir em requisitos não funcionais e em requisitos funcionais.

A identificação destes requisitos, para além de ter sido escrutinada com colaboradores da iTGROW e da CRITICAL Software, SA que acompanharam de perto o desenrolar deste estágio, foi igualmente acompanhada de um questionário, enviado a todos os colaboradores da área de engenharia destas empresas, tendo obtido o número significativo de 107 respostas, cuja análise se apresenta na secção seguinte.

4.1.1 Resultado das respostas ao questionário

Analisadas as 107 respostas dadas ao questionário apresentado no Anexo 2 e disponibilizado a todos os colaboradores da iTGROW e da CRITICAL Software, SA obtivemos os seguintes resultados.

Relativamente ao idioma mais utilizado nas pesquisa de informação empresarial, pode verificar-se, ver Gráfico 4, que a totalidade dos respondentes (107) utiliza a língua inglesa que pode ser complementada com a língua portuguesa (65). As restantes línguas não obtiveram um número de respostas significativo.

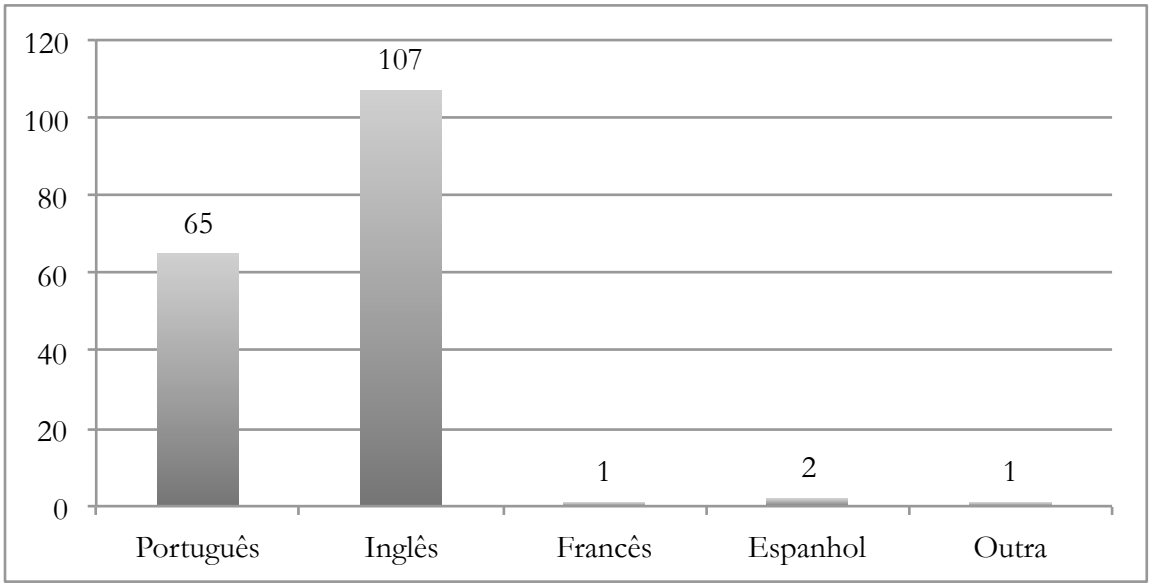


Gráfico 4 - Idioma utilizado com mais frequência na pesquisa de informação empresarial

No que respeita aos elementos considerados mais relevantes, na pesquisa de informação empresarial, a maioria dos respondentes (36) refere o tipo de documento. No entanto, há um número significativo de respondentes que refere o sumário (29) e o título (23) como elementos relevantes, ver Gráfico 5.

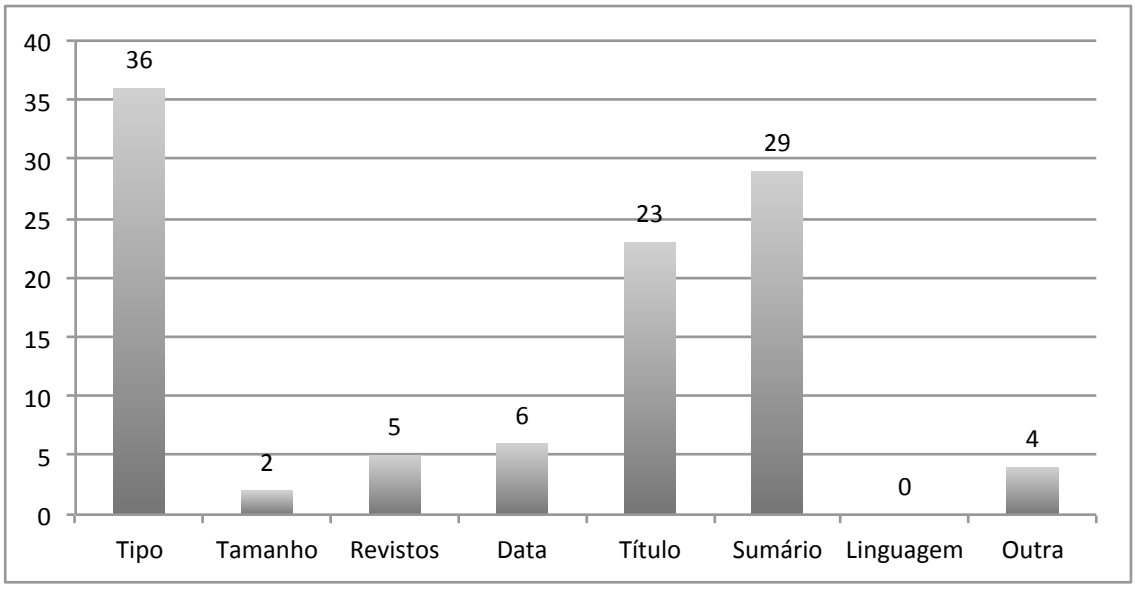


Gráfico 5 - Elemento considerado mais relevante na pesquisa de informação empresarial

Quanto ao tipo de filtro que os respondentes consideram mais relevante na pesquisa de informação empresarial, a grande maioria (70) refere a *keyword*, no entanto um número significativo refere o tipo de documento (30), ver Gráfico 6.

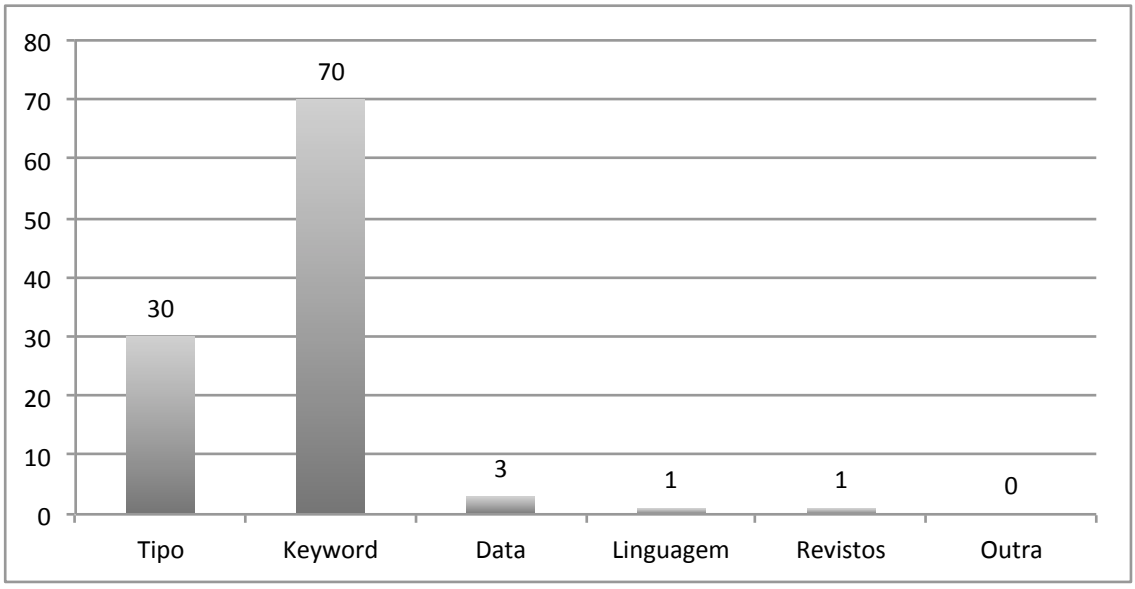


Gráfico 6 - Tipo de filtro considerado mais relevante na pesquisa de informação empresarial

Também a relevância dos resultados obtidos foi tida em atenção, questionando-se os respondentes sobre qual ou quais dos seguintes critérios: completude; proximidade; densidade e/ou metadados, pode ser uma mais valia no âmbito da pesquisa de informação empresarial. A maioria dos respondentes (40) refere a proximidade como mais valia no âmbito da pesquisa de informação empresarial. No entanto também há um número de respostas significativo para cada um dos outro critérios (25) completude, (24) metadados e (13) densidade, ver Gráfico 7.

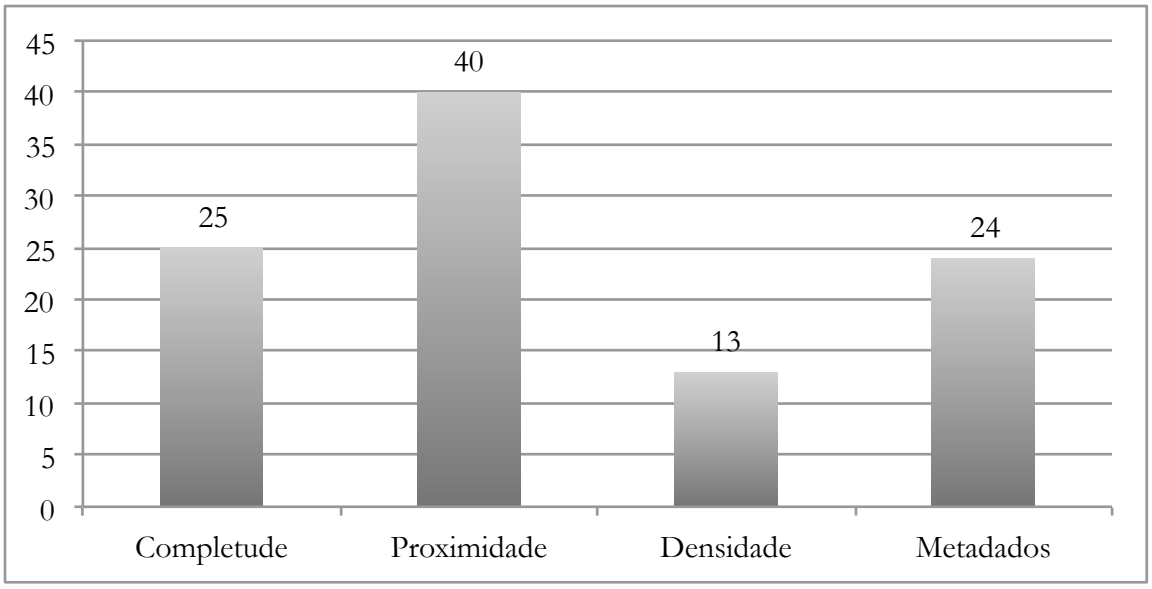


Gráfico 7 - Critérios que podem ser uma mais valia no âmbito da pesquisa de informação empresarial

Foram igualmente questionados os respondentes sobre qual o tipo de informação usada mais frequentemente como critério de pesquisa. A maioria dos respondentes (49) refere o uso de palavras-chave compostas como critério mais frequente na pesquisa de informação empresarial. No entanto, também deve ser tido em atenção o número significativo de respostas que refere a linguagem natural (28), assim como o critério das palavras-chave simples (24), ver Gráfico 8.

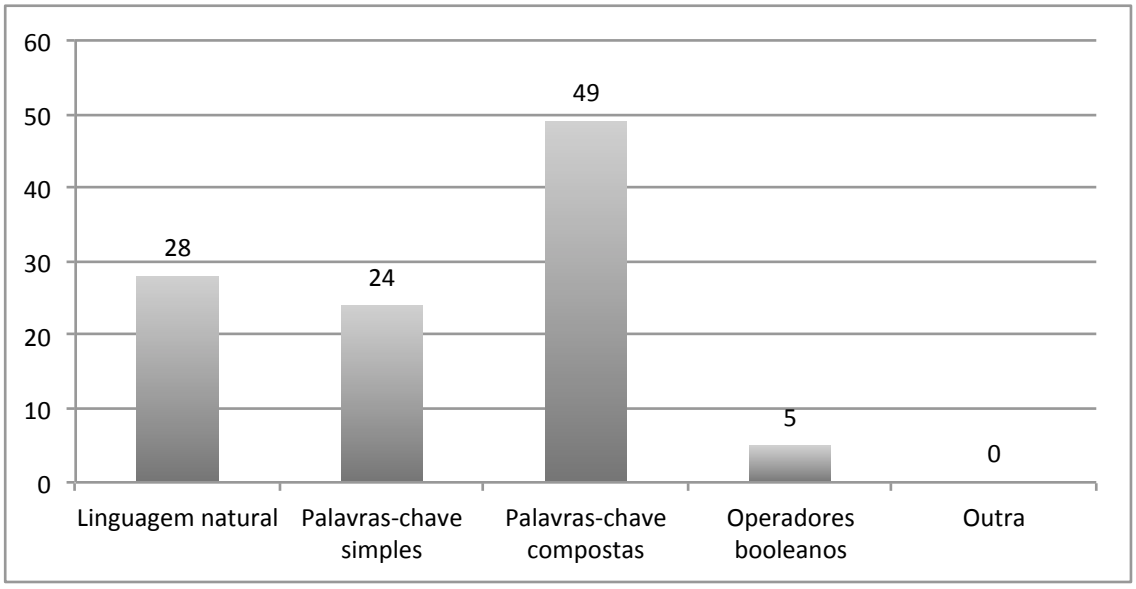


Gráfico 8 - Tipo de informação usada como critério de pesquisa de informação empresarial

Relativamente ao tipo de documentos mais frequentemente procurados os respondentes referem, com grande predominância (85), que os documentos que normalmente procuram são documentos com formato pdf, o formato doc(x) é indicado por (14) respondentes, ver Gráfico 9.

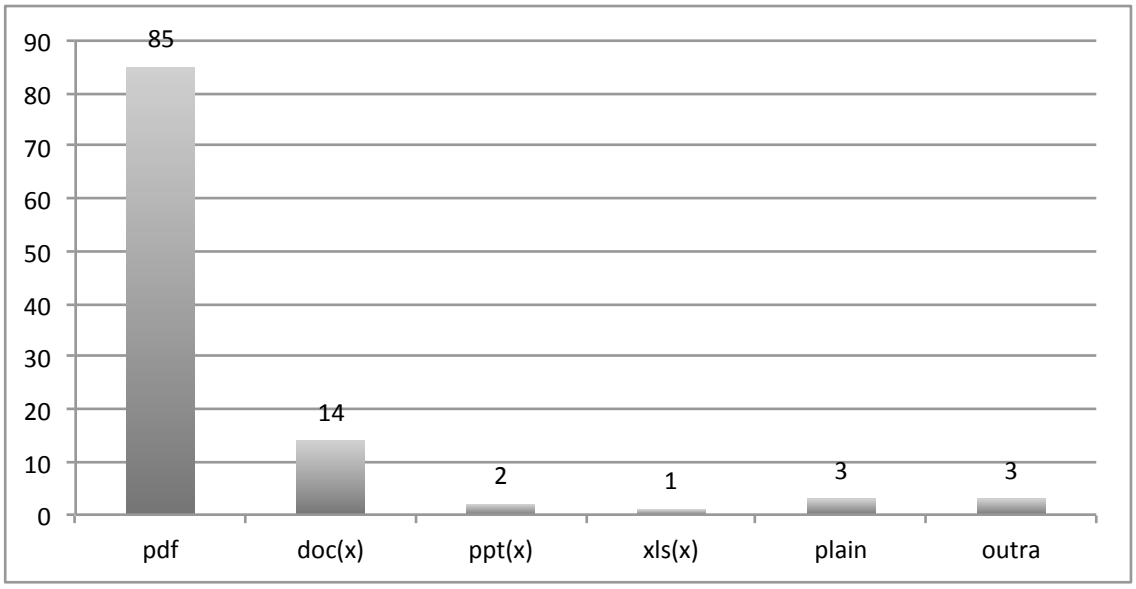


Gráfico 9 - Tipo de documentos mais procurados no âmbito da pesquisa de informação empresarial

Foram ainda questionados os respondentes, a este questionário, sobre a importância que dão à utilização de um mecanismo que sugira termos de pesquisa (sugestão automática) em função dos caracteres previamente digitados. A maioria dos respondentes (58), numa escala de 0 (pouco relevante) a 5 (muito relevante), refere o número 4, havendo (20) que referem igualmente o número 3 e o número 5 da referida escala, ver Gráfico 10. Portanto, os respondentes consideram relevante à existência de um mecanismo com estas características.

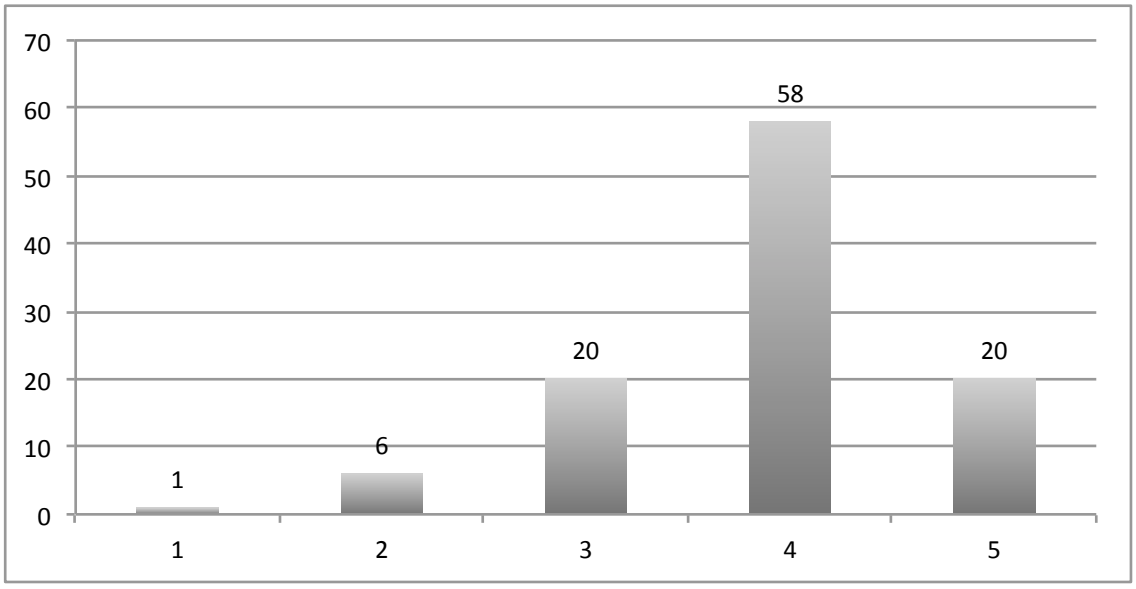


Gráfico 10 - Importância de um mecanismo de apoio à pesquisa de informação empresarial

Finalmente, os respondentes foram questionados sobre a relevância atribuída à antiguidade dos documentos a pesquisar. Grande parte dos respondentes (51) considere a antiguidade dos documentos como critério relevante, também aqui existem (23) que optam pelo número 3 da escala e (20) que optam pelo número 5 da escala, ver Gráfico 11.

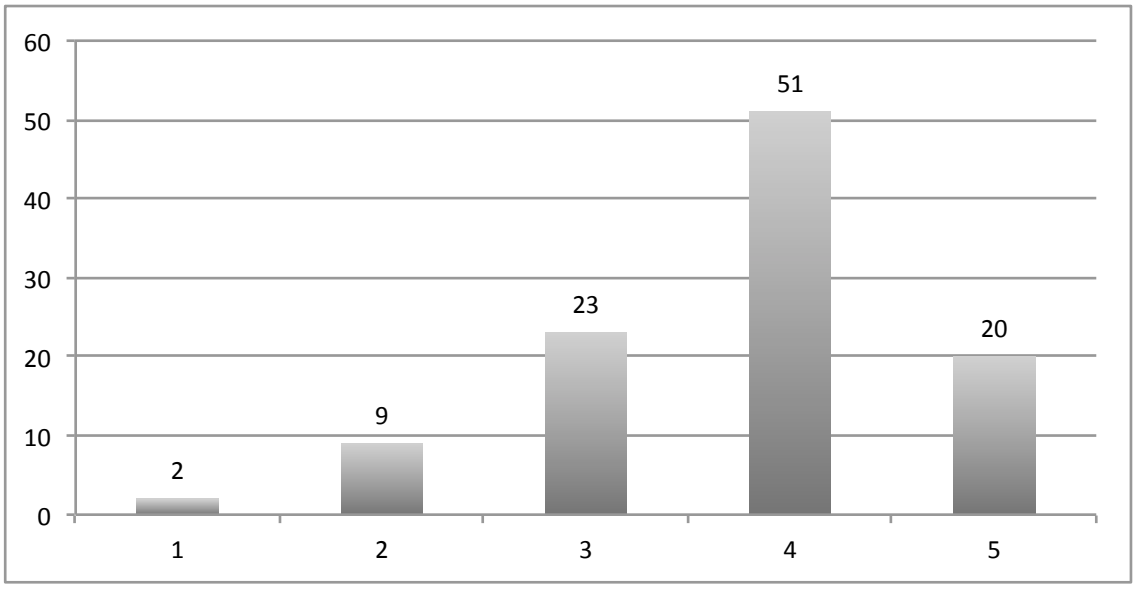


Gráfico 11 - Relevância atribuída à antiguidade dos documentos a procurar

Resumidamente, a análise das respostas ao inquérito, permite concluir que: o idioma mais utilizado é a língua inglesa; o elemento considerado mais relevante é tipo de documento, sendo também o tipo o filtro mais utilizado; a proximidade é um critério considerado como mais valia; a utilização de palavras-chave compostas é o critério de pesquisa considerado mais utilizado e o formato pdf é aquele que é mais frequentemente pesquisado.

Pode ainda referir-se que os respondentes dão muita importância à existência de um mecanismo de apoio à pesquisa, assim como dão grande relevância à antiguidade dos documentos a procurar.

4.1.2 Requisitos não funcionais

De acordo com Germoglio (2010) um dos objetivos de uma arquitetura de software é dar resposta aos requisitos não funcionais e funcionais. Os primeiros, normalmente, descrevem a forma que o sistema encontra para realizar as suas funções. Os segundos descrevem aquilo que o sistema faz. A par da resposta que deve ser dada aos requisitos não funcionais e funcionais a arquitetura deve obedecer a um determinado conjunto de atributos de qualidade concordantes com aqueles.

Para que se consiga dar resposta aos requisitos e aos atributos de qualidade é necessário um conhecimento rigoroso do que se pretende com o sistema a implementar e de forma interativa ir corrigindo a arquitetura para conseguir o pretendido, tendo em atenção as tecnologias utilizadas no desenvolvimento do projeto (Germoglio, 2010).

Deve ser dada particular atenção à existência de requisitos que possam, eventualmente, entrar em conflito. Estes conflitos entre requisitos têm impacto negativo nos atributos de qualidade dificultando a construção de uma arquitetura de qualidade (Germoglio, 2010).

Em função do modelo de qualidade podem existir atributos de qualidade distintos. De acordo com a norma ISO/IEC 9126-1:2001 são considerados atributos de qualidade os seguintes: funcionalidade; confiabilidade; usabilidade, eficiência; manutenibilidade e portabilidade. Barbacci *et al* (1995) consideram no seu relatório os seguintes atributos de qualidade: performance; confiança segurança e prevenção de acidentes. Por sua vez a Microsoft, (Meier et al., 2008), propõe uma *framework* que considera os seguintes atributos de qualidade: disponibilidade; integridade conceptual; interoperabilidade; manutenibilidade;

capacidade de ser gerível; performance; confiabilidade; reusabilidade; escalabilidade; segurança; suportabilidade; testabilidade e usabilidade.

Conforme podemos verificar, através dos exemplos apresentados anteriormente o contexto condiciona os atributos de qualidade definidos para garantir que a arquitetura de um determinado sistema satisfaz com os requisitos mínimos de qualidade. No entanto o modelo proposto pela norma ISO/IEC 9126-1:2001 parece ser consensual e atender a um conjunto de atributos de qualidade que garante que uma arquitetura desenvolvida tendo em vista o seu cumprimento satisfaz os requisitos mínimos de qualidade.

Os requisitos não funcionais, normalmente aplicam-se ao sistema e não a componentes do sistema. No âmbito de um sistema empresarial de pesquisa, conforme apresentado anteriormente, podemos destacar pela sua relevância os seguintes requisitos não funcionais:

- performance: o sistema deverá, devolver os resultados de cada pesquisa num espaço de tempo inferior a 5 segundos, pelo menos para 100 utilizadores concorrentes;
- segurança: o sistema deverá devolver informação com base na categoria do utilizador;
- extensibilidade: o sistema deverá aceitar novos módulos (p. ex. *crawler* para outro tipo de repositório), sem precisar de *downtime*, desde que os novos módulos respeitem a mesma estrutura base.

Relativamente ao requisito extensibilidade, foi necessário criar três módulos principais para o *crawler*, *parser* e *indexer*. Estes módulos irão carregar as novas funcionalidades enquanto o sistema estiver em funcionamento e usarão a referida funcionalidade com base no tipo de repositório.

O cumprimento do requisito performance leva a que, quando o utilizador realize uma pesquisa através da interface *Web*, é enviado um pedido ao Apache Solr que devolve um número limitado de elementos em vez de devolver todos os elementos encontrados, sendo os resultados paginados, quando o utilizador muda de página o Apache Solr carrega o próximo conjunto de resultados.

Por fim para cumprir o requisito relativo à segurança, é necessário proceder à indexação das permissões de cada ficheiro para garantir que apenas os utilizadores autorizados tenham acesso a essa informação.

4.1.3 Requisitos funcionais

No âmbito dos requisitos funcionais de um sistema empresarial de pesquisa devem ser considerados dois grandes grupos de requisitos que o sistema deverá cumprir e que designaremos de macro requisitos:

- o sistema deverá ter capacidade para procurar, encontrar, recolher, distribuir e indexar informação armazenada em repositórios heterogéneos (*SVN* e *AsknowledgeDb*), não descurando os permissões de cada um dos utilizadores do sistema (SR - 1.0.0[20]);
- os utilizadores deverão conseguir interagir com o sistema para realizarem as suas pesquisas com recurso a uma interface *user friendly* (SR - 2.0.0).

O primeiro macro requisito (SR - 1.0.0), indica que o sistema deverá ser capaz de procurar, encontrar, recolher, distribuir e indexar informação em dois tipos de repositórios (*file based* e *database record based*). Consequentemente, o sistema terá capacidade para procurar e recolher documentos que posteriormente serão indexados, não esquecendo os mecanismos

[20] na notação SR – X.Y.Z, SR, é relativo a "*System Requirement*"; X, é relativo ao macro requisito, Y, é relativo ao requisito e Z, é relativo a requisitos dependentes

associados à ACL. O sistema de procura deverá ainda ser capaz de recolher informação que permita identificar/caracterizar o documento (*metadados*) e, por fim, deverá indexar os documentos recolhidos no motor de pesquisa possibilitando aos utilizadores a consulta desses mesmos documentos.

Assim, no âmbito deste macro requisito, apresentamos de seguida de forma detalhada um conjunto de requisitos funcionais com as respetivas dependências, com recurso ao modelo utilizado por (Barrigas, 2014).

Identificação do repositório

Considerando que vamos procurar informação em dois repositórios diferentes (repositório *SVN* e repositório *AsknowledgeDb*) devemos proceder, inicialmente, à identificação do repositório em que se vai processar a procura de informação, obedecendo este processo a alguns requisitos funcionais, especificados na Tabela 9, na Tabela 10, e na Tabela 11.

A Tabela 9 mostra o requisito que permite identificar o tipo de repositório, a utilizar na procura de informação, com base na sua estrutura específica.

SR - 1.01.00 – Identificação do repositório							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
Um repositório terá uma estrutura específica com base no seu tipo (*file based* ou *database record based*) e poderá armazenar vários documentos com diferentes *MIME types*							

Tabela 9 - Requisito funcional (SR - 1.01.00 - identificação do repositório)

O requisito, apresentado na Tabela 10 "SR - 01.01.01 - Repositório *file based*" depende do requisito "SR - 1.01.01 - Identificação do repositório" e é selecionado se o repositório for identificado com um repositório *file based*.

SR - 1.01.01 - Repositório *file based*							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.01.00
Um repositório *file based* será composto por uma estrutura semelhante a um sistema de ficheiros onde existirá uma raiz e várias ramificações e ficheiros							

Tabela 10 - Requisito funcional (SR - 1.01.01 - repositório *file based*)

O requisito, apresentado na Tabela 11 "SR - 01.01.02 - Repositório *database record based*" depende, igualmente, do requisito "SR - 1.01.01 - Identificação do repositório" e é selecionado se o repositório for identificado com um repositório *database record based*.

SR - 1.01.02 - Repositório *database record based*							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.01.00
Um repositório *database record based* será composto por várias tabelas (colunas e linhas) onde os dados serão armazenados							

Tabela 11 - Requisito funcional (SR - 1.01.02 - repositório *database record based*)

Mecanismo de procura de informação no repositório

Conforme já referido anteriormente, devido à especificidade dos repositórios que serão acedidos por este protótipo, será desenvolvido um mecanismo de procura para cada um desses repositórios. Este importante componente deverá igualmente obedecer a alguns requisitos funcionais que destacamos nas tabelas seguintes.

A Tabela 12 mostra o requisito funcional relativo à especificidade de existir um mecanismo de procura próprio para cada um dos repositórios considerados no desenvolvimento e implementação deste protótipo.

SR - 1.02.00 - Mecanismo de procura							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
Devido às especificidades de cada um dos repositórios, *SVN* e *AsknowledgeDb*, deverá ser desenvolvido para cada um deles o seu próprio mecanismo de procura							

Tabela 12 - Requisito funcional (SR - 1.02.00 - mecanismo de procura)

O requisito, apresentados na Tabela 13, depende do requisito "SR - 1.02.00 - mecanismo de procura" e está relacionado com determinadas situações que possam ter ocorrido no estado do repositório, designadamente: inserção de novos documentos; atualização de documentos ou eliminação de documentos.

SR - 1.02.01 - Alteração do estado do repositório							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.00
O mecanismo de procura deverá ter em atenção situações relacionadas com uma possível alteração do estado do repositório, designadamente: adição de novos documentos; atualização de documentos ou exclusão de documentos							

Tabela 13 - Requisito funcional (SR - 1.02.01 - alteração do estado do repositório)

O requisito, apresentado na Tabela 14, depende do requisito "SR - 1.02.01 - alteração do estado do repositório", e está relacionado com a inserção de novos documentos no repositório.

SR - 1.02.01.01 - Adição de novos documentos ao repositório							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.01
O mecanismo de procura deverá ter capacidade para lidar com situações em que possa ter ocorrido a inserção de novos documentos no repositório							

Tabela 14 - Requisito funcional (SR - 1.02.01.01 - adição de novos documentos ao repositório)

O requisito, apresentado na Tabela 15, depende, igualmente, do requisito "SR - 1.02.01 - alteração do estado do repositório", e está relacionados com a eliminação de documentos do repositório.

SR - 1.02.01.02 - Exclusão de documentos do repositório							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.01
O mecanismo de procura deverá estar preparado para lidar com situações em possa ter ocorrido a exclusão de documentos do repositório							

Tabela 15 - Requisito funcional (SR - 1.02.01.02 - exclusão de documentos do repositório)

O requisito, apresentado na Tabela 16, depende, também, do requisito "SR - 1.02.01 - alteração do estado do repositório", e está relacionado com a atualização de documentos no repositório.

SR - 1.02.01.03 - Atualização de documentos do repositório							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.01
O mecanismo de procura deverá estar preparado para lidar com situações em que possa ter ocorrido a atualização de documentos no repositório							

Tabela 16 - Requisito funcional (SR - 1.02.01.03 - atualização de documentos no repositório)

O requisito apresentado na Tabela 17, depende do requisito "SR - 1.02.00 - mecanismo de procura", e está relacionado com a especificidade da procura por *MIME type*. Este requisito é reforçado pelos resultados do questionário apresentados na secção 4.1.1.

SR - 1.02.02 - Procurar documentos por *MIME type*							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.00
O mecanismo de procura deverá estar preparado para procurar documentos em função dos seguintes *MIME types*: pdf; word; powerpoint; texto simples ou registos de bases de dados							

Tabela 17 - Requisito funcional (SR - 1.02.02 - procurar documentos por *MIME type*)

O requisito apresentado na Tabela 18, depende, igualmente, do requisito "SR - 1.02.00 - mecanismo de procura", e está relacionado com a especificidade de ser necessário associar mecanismos da ACL à procura.

SR - 1.02.03 - Associar ACL à procura de documentos							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.00
O mecanismo de procura deverá estar preparado para associar a ACL de repositório a cada um dos documentos procurados para posterior indexação							

Tabela 18 - Requisito funcional (SR - 1.02.03 - associar ACL à procura de documentos)

O requisito apresentado na Tabela 19 também está dependente do requisito "SR - 1.02.00 - mecanismo de procura", e está relacionado com a especificidade associada à capacidade de extração de conteúdos. Este requisito é reforçado pelos resultados do questionário apresentados na secção 4.1.1.

SR - 1.02.04 - Gestão de documentos							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.00
Quando o mecanismo de procura encontra um documento, que deva ser indexado, deverá estar preparado para extrair e indexar a seguinte informação: *MIME type*; conteúdo do ficheiro; *tags*; localização; ACL; URI; data e ação							

Tabela 19 - Requisito funcional (SR - 1.02.04 - gestão de documentos)

O requisito, apresentado na Tabela 20, depende, também, do requisito "SR - 1.02.00 - mecanismo de procura", e está relacionado com as especificidades do output do mecanismo de procura.

SR - 1.02.05 - Output do mecanismo de procura							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.02.00
O mecanismo de procura deverá estar preparado para converter umaa mensagem para formato POJO (*Plain Old Java Object*) e publicá-lo numa fila							

Tabela 20 - Requisito funcional (SR - 1.02.05 - output do mecanismo de procura)

Gestão de mensagens

Os documentos recolhidos pelo mecanismo de procura nos repositórios selecionados são enviados para um mecanismo que sujeita os documentos referidos às necessárias operações de transformação, através de uma fila de mensagens. Também aqui é necessário respeitar algumas regras, designadamente a capacidade que este mecanismo de ter para aceitar diversos formatos, ver Tabela 21.

SR - 1.03.00 - Fila de documentos para transformar							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
Deverá existir um mecanismo de fila entre o mecanismo de procura e o mecanismo de transformação de documentos que deverá estar preparado para aceitar documentos com qualquer tipo de formato (XML, POJO, etc.)							

Tabela 21 - Requisito funcional (SR - 1.03.00 - fila de documentos para transformar)

À semelhança do que se passa entre o mecanismo de procura e o mecanismo de transformação, também circularão mensagens entre o mecanismo de transformação e o mecanismo de indexação. Assim, os ficheiros tratados pelo mecanismo de transformação são enviados para uma fila de mensagens, sendo necessário respeitar algumas regras, designadamente a capacidade para aceitar diversos formatos, ver Tabela 22.

SR - 1.04.00 - Fila de documentos para indexar							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
Deverá existir um mecanismo de fila responsável pela gestão de mensagens entre o mecanismo de transformação e o mecanismo de indexação que deverá estar preparado para aceitar documentos com qualquer tipo de formato							

Tabela 22 - Requisito funcional (SR - 1.04.00 - fila de documentos para indexar)

Mecanismo de transformação de documentos

Foi desenvolvido um mecanismo de transformação cujo objetivo consiste em extrair e tratar informação dos documentos antes destes serem enviados para o mecanismo de indexação, cujo requisito é o que apresentamos na Tabela 23.

SR - 1.05.00 - Mecanismo de transformação							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
Deverá existir um mecanismo de transformação com capacidade para analisar documentos sujeitos a múltiplos *MIME types*							

Tabela 23 - Requisito funcional (SR - 1.05.00 - mecanismo de transformação)

O requisito, apresentados na Tabela 24, depende do requisito "SR – 1.05.00 – mecanismo de transformação" e tem o objetivo de extrair informação do conteúdo de um documento.

SR - 1.05.01 - Extração de informação do conteúdo de um documento							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.05.00
O mecanismo de transformação deverá estar preparado para extrair de um documento, sempre que possível, a seguinte informação: título; *abstract*; corpo do texto; autor; data; ACL; resumo; tipo de documento (memorando; SGQ, etc.)							

Tabela 24 - Requisito funcional (SR - 1.05.01 - extração de informação do conteúdo de um documento)

O requisito, apresentados na Tabela 25, depende, igualmente, do requisito "SR – 1.05.00 - mecanismo de transformação" e tem o objetivo criar *tags* de metadados para associar a determinado documento.

SR - 1.05.02 - Extração de metadados							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	SR - 1.05.00
O mecanismo de transformação deverá estar preparado para criar *tags* de metadados que serão associados a um determinado documento							

Tabela 25 - Requisito funcional (SR - 1.05.02 - extração de metadados)

Mecanismo de indexação de documentos

Os documentos remetidos para o mecanismo de indexação são tratados conforme a *tag* que lhes está associada: "A" para adição; "D" para exclusão e "M" para modificação. Com base na *tag* o ficheiro é adicionado, modificado ou removido do mecanismo de indexação. Também neste componente é necessário respeitar algumas regras que são especificadas de seguida através do requisito apresentado na Tabela 26.

SR-1.06.00 - Mecanismo de indexação							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
Deverá existir um mecanismo de indexação preparado para mapear os campos das mensagens e criar os índices dos respetivos documentos							

Tabela 26 - Requisito funcional (SR - 1.06.00 - mecanismo de indexação)

Interface Web

O segundo macro requisito (SR - 2.0.0), indica que os utilizadores deverão interagir com o sistema de modo a conseguir realizar pesquisas sobre os documentos que este indexou. Normalmente nesta fase o utilizador digitará um termo de procura (frase ou palavras que de algum modo o utilizador associa ao documento que pretende de procurar), numa *interface* adequada a um sistema empresarial de pesquisa. Aquele termo, digitado pelo utilizador será devidamente analisada, recorrendo a algoritmos de pesquisa que devolverão uma lista de resultados considerados mais relevante para os interesses do utilizador.

A interface Web deve obedecer a um conjunto de requisitos que são apresentados na Tabela 27, na Tabela 28, na Tabela 29, na Tabela 30, na Tabela 31 e na Tabela 32.

O requisito apresentado na Tabela 27, tem por objetivo garantir a execução de procura de um documento, digitando os termos da procura. Este requisito é reforçado pelos resultados do questionário apresentados na secção 4.1.1.

SR - 2.01.00 - Pesquisa de documentos							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
A interface *Web* deverá estar preparada para possibilitar a execução de pesquisas, por documento, digitando na caixa de pesquisa os termos a pesquisar							

Tabela 27 - Requisito funcional (SR - 2.01.00 - pesquisa de documentos)

O requisito apresentado na Tabela 28, tem por objetivo garantir a possibilidade de serem aplicados alguns tipos de filtro. Este requisito é igualmente reforçado pelos resultados do questionário apresentados na secção 4.1.1.

SR - 2.02.00 - Filtrar documentos							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
A interface *Web* deverá permitir que seja possível filtrar os resultados por autor, localização e *MIME type* usando o filtro adequados							

Tabela 28 - Requisito funcional (SR - 2.02.00 - filtrar documentos)

O requisito apresentado na Tabela 29, tem por objetivo garantir a possibilidade de serem percorridos todos os resultados da procura, através de um mecanismo de paginação.

SR - 2.03.00 - Paginação							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
A interface *Web* deverá permitir percorrer todos os resultados devolvidos pela pesquisa usando o menu de paginação colocado no fundo da página							

Tabela 29 - Requisito funcional (SR - 2.03.00 - paginação)

O requisito apresentado na Tabela 30, tem por objetivo garantir a possibilidade de que os resultados sejam mostrados em função da sua relevância. Este requisito é reforçado pelos resultados do questionário apresentados na secção 4.1.1.

SR - 2.04.00 - Relevância							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Topo	Funcional					**Dependências**	Nenhuma
A interface *Web* deverá permitir que em cada pesquisa executada sejam mostrados os resultados devolvidos ordenados em função da sua relevância, por exemplo, do mais relevante para o menos relevante, com base nos filtros aplicados							

Tabela 30 - Requisito funcional (SR - 2.04.00 - relevância)

O requisito apresentado na Tabela 31, tem por objetivo garantir a possibilidade do utilizador abandonar uma sessão de trabalho de forma segura.

SR - 2.05.00 - Logout							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
A interface *Web* deverá permitir que o utilizador termine a sua sessão de trabalho de forma segura, pressionando o botão de *logout*							

Tabela 31 - Requisito funcional (SR - 2.05.00 - *logout*)

O requisito apresentado na Tabela 32, tem por objetivo garantir a possibilidade de que o utilizador proceda à sua autenticação usando as suas credenciais.

SR - 2.06.00 - Login							
Versão	1.0	**Fase**	1.0	**Estado**	Proposto	**Data**	19/02/2016
Tipo	Funcional					**Dependências**	Nenhuma
A interface *Web* deverá permitir que o utilizador proceda à sua autenticação no sistema usando as suas credenciais							

Tabela 32 - Requisito funcional (SR - 2.06.00 - *login*)

Após identificação dos requisitos não funcionais e funcionais temos a resposta à questão 1 "que requisitos, não funcionais e funcionais devem ser implementados?".

4.2 Elementos arquiteturais

O processo de procura, recolha, indexação e disponibilização de informação é feito com a participação de vários blocos de construção (*building blocks*), designadamente: procura de informação; indexação de informação; recuperação de informação e análise de conteúdo. Estes blocos de construção são identificados e implementados na fase de desenvolvimento do sistema de pesquisa de informação empresarial em função dos requisitos não funcionais e funcionais, entendidos como adequados por parte da empresa. Conforme já foi referido no Capítulo 3, o protótipo do sistema empresarial de pesquisa desenvolvido e implementado deve ter capacidade para recolher informação a partir de dois repositórios corporativos, em qualquer formato, transformando-a em informação estruturada que pode ser diretamente pesquisada e consultada (Benghozi & Chamaret, 2010).

Normalmente uma arquitetura é concebida para garantir que um sistema satisfaça os requisitos ou objetivos que lhe são associados. No entanto, independentemente da metodologia utilizada para identificação e especificação dos requisitos não funcionais e funcionais de qualquer sistema de pesquisa de informação empresarial, podem ser declarados dois objetivos que, pela sua generalidade e independentemente de outros poderem ser considerados, estão sempre associados a estes sistemas e que são:

- (1) eficácia (qualidade): capacidade para recuperar um conjunto de documentos, relevante, em resposta a uma qualquer consulta;
- (2) eficiência (velocidade): capacidade para responder às consultas que lhe são submetidas no mais curto espaço de tempo (Croft et al., 2015).

Assim, a arquitetura de um sistema de pesquisa de informação empresarial é fortemente determinada por estes dois requisitos (independentemente da identificação de outros). Isto acontece, porque o objetivo principal consiste em desenvolver um sistema eficiente que devolva resultados de alta qualidade. Para conseguirmos implementar um sistema que obedeça àqueles dois requisitos, não inviabilizando o atendimento de outros requisitos identificados, recorreremos aos *building blocks* (Croft et al., 2015).

Segundo Koren (2013) os sistemas de *information retrieval* baseiam-se na arquitetura que apresentamos na Figura 13, adaptada do trabalho referido, e que pode ser adaptada aos requisitos não funcionais e funcionais, específicos do sistema a ser implementado. Como podemos verificar na referida figura, os motores de busca de texto completo compreendem várias etapas de modo a fornecer ao utilizador os resultados adequados a uma determinada consulta.

A pesquisa, neste caso, é feita recorrendo a uma frase ou palavras-chave, opcionalmente enriquecida por operadores especiais que servem para refinar a consulta. É o utilizador do sistema *information retrieval* que define a consulta, esperando que o sistema cumpra, devolvendo os resultados adequados à consulta, satisfazendo assim a necessidade de informação do utilizador.

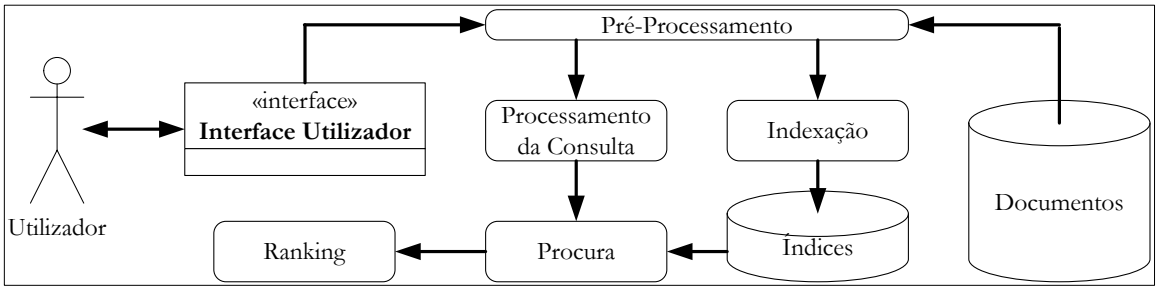


Figura 13 - Arquitetura geral de um sistema de pesquisa de informação (Koren, 2013)

Considerando o que já foi dito anteriormente e pensando no desenvolvimento do projeto para além do objetivo deste estágio, deve ser dada a devida atenção ao paradigma dos *bigdata*. Assim, propõe-se que a arquitetura apresentada na Figura 13, seja alterada em função do que é dito em relação a arquiteturas que possam suportar aquele paradigma pela Search Technologies.

Tendo, então, presente o que é sugerido pela Search Technologies é considerado um modelo de arquitetura para pesquisa de *bigdata* que se deve centrar no processamento do conteúdo da informação, antes de proceder à sua indexação, como mostrado na Figura 14 adaptada de (Search Technologies, 2015a).

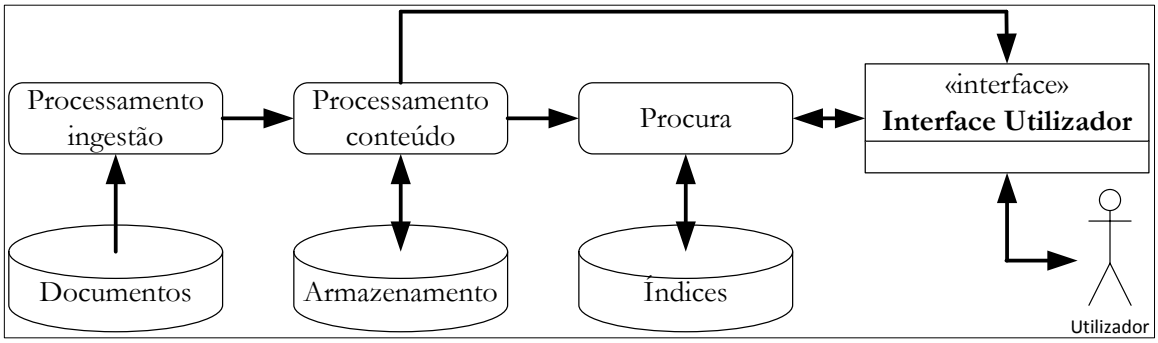


Figura 14 - Arquitetura de um sistema de pesquisa de informação com *bigdata* (Search Technologies, 2015a)

Numa visão de alto nível devem ter-se em atenção os aspetos relacionados com os blocos construtores em que pode ser considerado que os componentes de um motor de procura apoiam essencialmente as duas funções principais daquele motor, a saber: (1) o processo de indexação, mostrado na Figura 15 e (2) o processo de procura mostrado na Figura 16, ambas adaptadas de (Croft et al., 2015). O processo de indexação tem por objetivo a construção das estruturas que permitem que seja feita a procura, que usa essas estruturas, e seja devolvido um adequado conjunto de resultados (Croft et al., 2015).

Os principais componentes no processo de indexação, que podemos ver na Figura 15, são a aquisição de texto, a transformação de texto e a criação do índice.

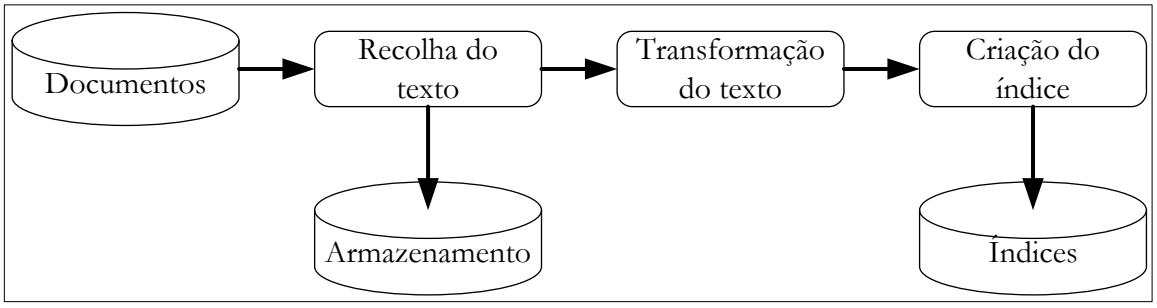


Figura 15 - Processo de indexação de um motor de busca (Croft et al., 2015)

Na Figura 16, podemos ver, igualmente, os principais *building blocks* do processo de procura de informação que são, designadamente: interação com o utilizador, *ranking* e avaliação.

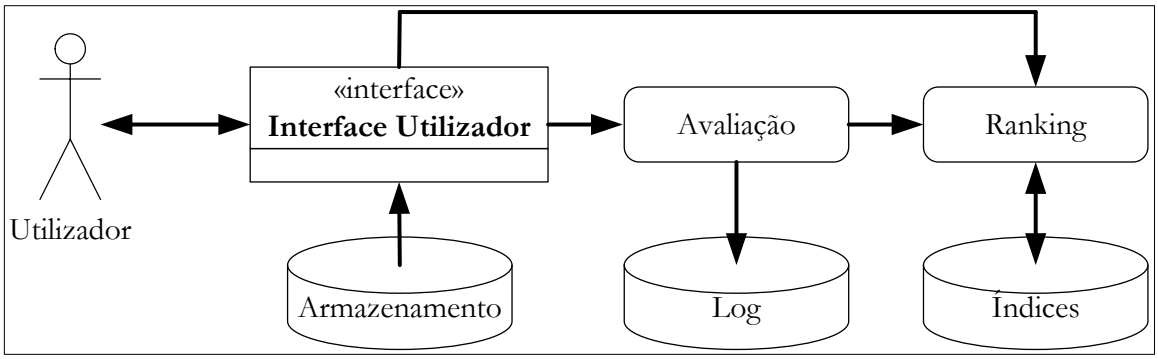


Figura 16 - Processo de procura de um motor de busca (Croft et al., 2015)

Assim, com base nas propostas de arquitetura, apresentadas nas figuras anteriores podemos adotar como arquitetura geral para um sistema de pesquisa de informação empresarial a arquitetura que é mostrada na Figura 17.

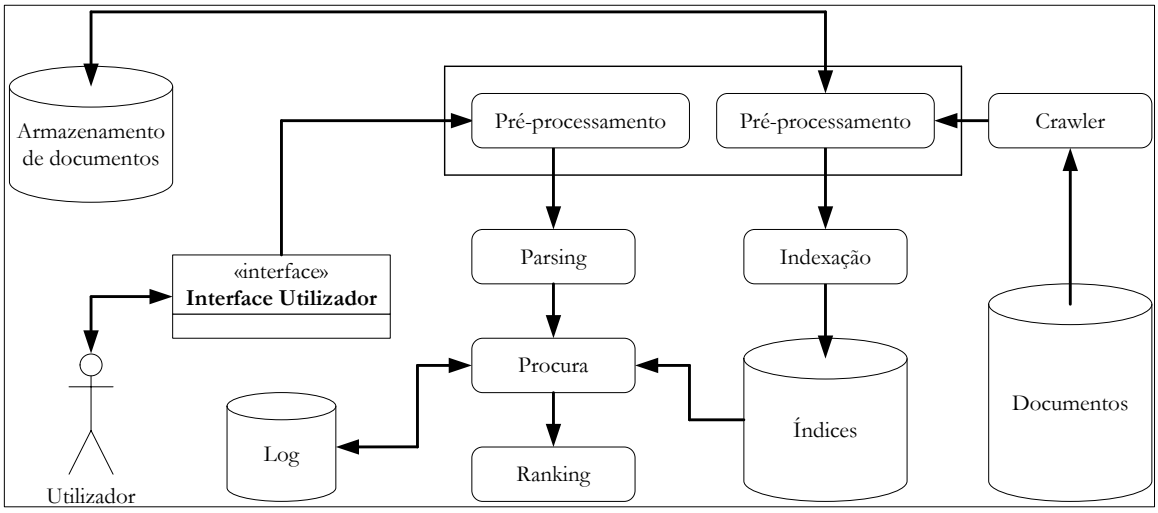


Figura 17 - Arquitetura geral de um sistema de pesquisa de informação empresarial

A Figura 17, que resulta da análise de vários estudos (Croft et al., 2015; Koren, 2013; Search Technologies, 2015a) dá resposta às questões 4 e 5, respetivamente "a arquitetura e as tecnologias propostas garantem a extensibilidade do sistema?" e "a arquitetura e as tecnologias propostas garantem que as questões relativas à qualidade e segurança da informação são acauteladas?" apresentadas no Capítulo 3, secção 3.1.

4.3 Building blocks

Conforme poderemos perceber, pelo que fica dito anteriormente os *building blocks* são essenciais para a construção de qualquer sistema de pesquisa de informação. Estes componentes podem ser divididos em dois grandes grupos: (1) gerais e (2) diferenciadores.

4.3.1 Building blocks gerais

De seguida identificamos um conjunto de *building blocks* que normalmente estão associados ao desenvolvimento de um sistema de pesquisa de informação empresarial (White, 2012):

- ***Content gathering*** - este componente tem por função procurar e reunir a informação que será indexada. Devem estar presentes algumas preocupações relativamente a este componente que poderão influenciar a sua implementação, designadamente: que tipo de informação deve ser pesquisada e indexada; possibilidade de existir informação que

não é encontrada ou que é passada incorretamente para o indexador e ainda que a localização da informação pode mudar.

- ***Connectores*** - na pesquisa de informação empresarial a informação que será indexada encontra-se em diferentes repositórios. Portanto, os *connectors* são componentes essenciais que serão usados como interface entre dois ou mais componentes que não tenham sido originalmente projetados para funcionarem de forma integrada.
- ***Document filtering*** - a informação procurada, só muito raramente estará num formato *plain text*. Normalmente, esta informação encontra-se em formatos diversos, tais como, por exemplo pdf, word, powerpoint, etc.. Assim, cada um destes formatos deverá ser convertido para *plain text* com recurso a estes componentes tecnológicos.
- ***Parsing and tokenising*** - processo em que o texto é separado em elementos distintos: palavras e frases (*tokens*), estes são separados usando espaços ou caracteres de pontuação. Este componente deverá estar habilitado para identificar e lidar com acrónimos, o que significa que, em muitos casos, o *tokenizer* terá que identificar a linguagem utilizada.
- ***Stop words*** - em qualquer linguagem existem palavras que não têm valor como termos de pesquisa. Por exemplo, em inglês as palavras *the*, *of* e *about*. No entanto, devemos ter presente que, em ambiente empresarial, certas palavras ou elementos de pontuação podem ser usadas em nomes de projetos ou produtos.
- ***Stemming* e *lemmatization*** - a plasticidade de qualquer linguagem natural está associada à grande variedade de formas que podem ser utilizadas para transmitir a mesma ideia. Esta característica da linguagem natural é um problema para os motores de pesquisa que funcionam comparando os termos de procura com os termos encontrados nos documentos. Para lidar com este problema têm sido desenvolvidas várias abordagens no âmbito deste componente *stemming* e *lemmatization*.
- **Construção e manutenção do índex** - genericamente o índex pode ser construído e mantido de duas formas: uma que procura encontrar ocorrências de um determinado padrão, técnica apropriada quando o texto é pequeno e sendo a única escolha quando o texto é muito volátil; uma outra que envolve a criação de índices sobre o texto de modo a aumentar a velocidade de procura, opção mais adequada quando existe uma grande quantidade de texto semi-estático. Devemos ainda considerar que existem três técnicas de indexação: (1) *inverted files,* rápida na pesquisa mas mais lenta na indexação ; (2) *suffix arrays,* mais rápida para procurar frases e outras *queries* menos comuns, mas mais complicada de manter e (3) *signature files,* esta técnica apresenta pior desempenho que a técnica *inverted files*.
- **Segurança e lista de controlo de acessos** - existem duas técnicas que podem ser usadas para gerir o acesso à informação: (1) a primeira técnica, baseada na ACL, permite que o utilizador apenas veja os resultados das categorias a que tem acesso, esta técnica pressupõe a existência de categorias específicas com acesso limitado e tem um impacto reduzido na velocidade de recuperação de informação, sendo normalmente designada *early-binding*; (2) a segunda técnica permite que o utilizador faça a pesquisa desejada devendo, antes dos resultados serem mostrados, proceder à sua autenticação, serão visualizados apenas os resultados a que o utilizador tem permissão para aceder, esta técnica é normalmente designada *late-binding*.
- **Gestão da pesquisa** - o processo de gestão da pesquisa constitui-se como um processo no qual o sistema transforma a pesquisa do utilizador de forma que o motor de pesquisa a consiga utilizar *query binding*. Podemos destacar os seguintes tipos de pesquisas: boleanas; linguagem natural; *thesaurus queries*; *fuzzy queries*; termos de pesquisa e probabilísticas.

- *Spell checking* - este componente tem por objetivo detetar erros ortográficos e oferecer sugestões para que os mesmos possam ser corrigidos.
- *Retrieval models* - este componente, apresenta quatro modelos diferentes:
  - *boolean retrieval*, neste modelo podemos identificar as seguintes características: as *queries* são especificadas como expressões booleanas com semânticas precisas; os resultados são devolvidos com base num critério de decisão binário, sem noção de peso nos termos; complexidade em traduzir a informação necessária em expressões boleanas e modelo dominante em bases de dados de pesquisa de documentos;
  - *vector space model*, neste modelo é reconhecido que o uso de pesos binários é demasiado limitador e que o *matching* parcial é possível, atribuindo pesos não binários nos termos das *queries* e nos documentos que são posteriormente usados para determinar o grau de similitude e ainda que podem ser encontradas as seguintes desvantagens: a ordem em que os termos aparecem nos documentos é perdida; os documentos com contexto similar mas termos diferentes não são considerados; os termos têm que ter um *match* exato;
  - *probability theory model*, neste modelo a ideia consiste em criar um conjunto de documentos, que apenas contém os documentos que sejam considerados relevantes, com base na *query* usada.
  - *latent semanting indexing*, o objetivo neste modelo é associar documentos e *queries* a determinados conceitos, usando técnicas matemáticas para identificar padrões entre termos e conceitos.
- *Ranking* - na determinação do ranking, idealmente, os resultados deverão ser ordenados de forma descendente tendo em atenção a sua relevância. Existem duas principais técnicas para proceder à ordenação por relevância: *relative query boosting* e *absolute query boosting*.
- *Summarization* - a lista de resultados normalmente é apresentada por um título e informação adicional sobre o documento (metadados) e o seu sumário.

4.3.2 Building blocks diferenciadores

Para além dos componentes identificados anteriormente, podemos referir um outro conjunto de *building blocks* que podem funcionar como fatores diferenciadores do sistema de pesquisa. No âmbito desta vertente diferenciadora, podemos destacar os seguintes componentes:

- *Entity extraction* - o conceito *entity extraction*, está relacionado com a capacidade da aplicação de procura conseguir identificar, automaticamente, certos termos como sendo relevantes para indexar, sem que os mesmos tenham que ser indexados manualmente, por exemplo: identificar nomes de pessoas; identificar empresas; identificar datas.
- *People search* - aspeto considerado de grande importância na extração de entidades, consiste na pesquisa de pessoas por nome ou especialização.
- *Federated search* - normalmente a informação encontra-se dividida em vários repositórios, no entanto o utilizador não deverá saber onde está a informação que a aplicação deve encontrar. Existem duas formas de realizar este tipo de pesquisa: (1) a primeira consiste em construir um *index* para todo o conteúdo de cada repositório, em teoria esta seria a melhor maneira mas na realidade o tamanho do *index* pode ser significativo; (2) a segunda forma passa pela existência de um mecanismo de pesquisa em cada repositório que devolva o resultado à aplicação principal, esta forma de pesquisa pode trazer problemas relativamente à relevância dos resultados.
- *Duplicate and similar documents* - a duplicação de documentos, ou documentos quase idênticos pode ocorrer em várias situações, esta situação pode ser ultrapassada

utilizando algoritmos para detetar documentos duplicados para que não sejam indexados, para detetar documentos exatos podem ser usadas técnicas de *checksumming* e para detetar documentos similares podem ser usados algoritmos de *fingerprinting*.

- ***Mobile search*** - com a rápida adopção de *smartphones* e *tablets* pela população, um dos novos desafios é apresentar a informação de forma coerente nesses dispositivos, sendo expectável que a pesquisa se adapte ao contexto, com base na localização do utilizador, podendo os resultados serem diferentes.
- ***Faceted search*** - este tipo de pesquisa envolve a utilização de vários filtros (*facets*) à *query* de modo a realizar *drill down*, no que inicialmente seria um elevado número de resultados à *query* escolhida.
- ***Multilingual search*** - no caso da informação se encontrar em várias línguas existem duas opções: (1) o utilizador insere os termos de procura na linguagem apropriada ou (2) a aplicação usa um gerador de sinónimos, pelo que a aplicação realiza a mesma pesquisa em linguagens diferentes.
- ***Semantic search*** - recentemente tem havido um grande esforço no sentido de melhorar a pesquisa fazendo com que o motor de pesquisa consiga perceber o contexto dos termos de pesquisa, este esforço ainda se encontra numa fase de desenvolvimento e pesquisa.

4.4 Arquitetura do protótipo

Com base na identificação e especificação dos requisitos não funcionais e funcionais e também no estudo feito sobre elementos arquiteturais associadas à pesquisa de informação empresarial e na análise dos *building blocks* apresentados anteriormente, pode ser apresentada uma arquitetura que servirá de base ao desenvolvimento e implementação do protótipo relativo ao sistema CYCLOPS. No essencial o protótipo será constituído por cinco grandes blocos, ver Figura 18: (1) mecanismo de procura; (2) mecanismo de transformação; (3) mecanismo de indexação; (4) mecanismo de gestão de mensagens e (5) interface *Web*.

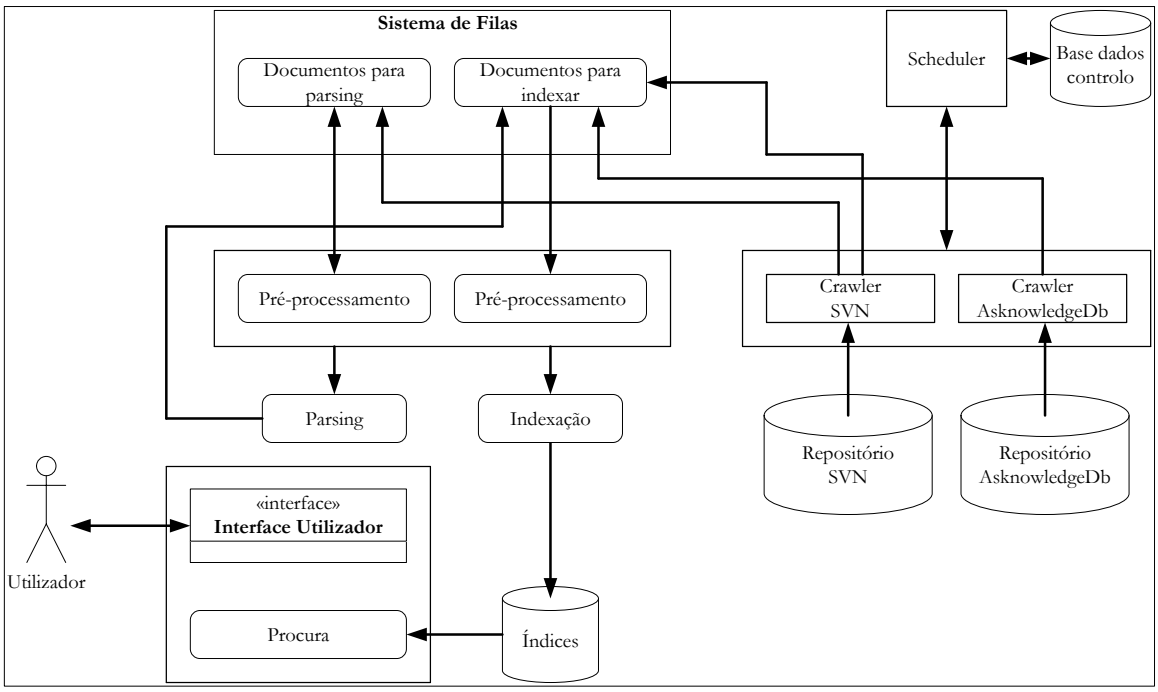


Figura 18 - Arquitetura do protótipo do sistema CYCLOPS

Com base nos elementos arquiteturais, na identificação dos requisitos não funcionais e funcionais e tendo em atenção os atributos de qualidade baseados na norma ISO/IEC 9126-1:200, é proposta uma arquitetura, para desenvolvimento do protótipo CYCLOPS, apresentada na Figura 18.

Estão, agora, respondidas as questões 2 e 3, respetivamente: "que arquitetura deve ser utilizada para responder de forma adequada ao pretendido?" e "que tecnologias são adequadas à implementação dos requisitos identificados?".

Capítulo 5
Desenvolvimento do Protótipo

O desenvolvimento do protótipo, como prova de conceito, associado ao sistema CYCLOPS é feito em conformidade com a arquitetura apresentada no capítulo anterior, Figura 18. Assim, nas secções seguintes, para além de algumas considerações sobre o servidor de pesquisa empresarial, apresentamos o que foi feito para desenvolver e implementar cada um dos cinco grandes blocos referidos relativamente à arquitetura referida anteriormente: (1) mecanismo de procura; (2) mecanismo de transformação; (3) mecanismo de indexação; (4) mecanismo de gestão de mensagens e (5) interface *Web*.

5.1 Servidor de pesquisa empresarial

Conforme referido no Capítulo 2 há estudos que apontam no sentido da utilização de motores de pesquisa *open source*. Como razões relevantes para esta aposta destacam-se a simplificação do desenvolvimento e a minimização dos custos de desenvolvimento (O'Riordan, 2014; Valentín-Rodríguez et al., 2015). Assim, optamos por utilizar no desenvolvimento e implementação deste protótipo tecnologias *open source*.

Dos sistemas empresariais de pesquisa *open source*, optamos por utilizar a plataforma Apache Solr. Esta plataforma é um servidor de pesquisa empresarial que utiliza o Apache Lucene, biblioteca de pesquisa de alto desempenho baseada em Java, para disponibilizar várias funcionalidades em HTTP. Esta opção é suportada em várias análises comparativas com destaque para os estudos de Middleton e Baeza-Yates (2011; 2007), que concluem, respetivamente, que o Apache Lucene é uma boa alternativa, como sistema empresarial de pesquisa, considerando que: utiliza pouco espaço de armazenamento; tem um baixo tempo de recuperação e é um sistema de pesquisa bastante competitivo considerando a utilização que faz da memória e o seu desempenho em tempo de pesquisa.

Um sistema empresarial de pesquisa, em termos de arquitetura, pode ser descrito em função daqueles que serão os seus componentes essenciais: *crawler*; *parser*; *indexer, searcher* e interface *Web*. O *crawler* tem a função de procurar e recolher novos documentos e proceder ao seu envio para o *parser* que tem como função identificar diferentes tipos de documentos. O *indexer* percorre os dados analisados pelo *parser* e identifica as palavras-chave construindo um índice que indica em que termos essas palavras são usadas nos documentos. Finalmente, o *searcher* recorre ao índice para dar resposta às consultas colocadas pelos seus utilizadores (Molková, 2011; Wallé, 2008), ver Figura 19.

O *indexer* e o *searcher* são dois componentes disponibilizado pela plataforma de pesquisa empresarial Apache Solr. São estes componentes que permitem que o sistema empresarial de pesquisa, rapidamente, recupere os resultados em função da consulta feita pelo utilizador. Estes componentes podem ser configurados através de ficheiros de configuração XML. Os restantes componentes que compõem o sistema empresarial de pesquisa (*crwaler*, *parser* e

interface) foram desenvolvidos em linguagem de programação Java 8. A opção por esta linguagem tem em atenção que o Apache Lucene tem fortes ligações com esta linguagem de programação, ver Figura 19.

A interligação entre os vários componentes que constituem o sistema empresarial de pesquisa é feita através de um sistema de filas que recebe os documentos de um determinado produtor e os disponibiliza a qualquer consumidor que deles necessite, sendo assim responsável pela sua gestão. Para este mecanismo de recepção e distribuição de documentos foi utilizada a plataforma Apache Kafka. Esta plataforma configura um sistema de mensagens distribuído de alto desempenho suportando distribuição e processamento em tempo real que, de facto, se adapta a um mecanismo de distribuição e transporte capaz de transportar e distribuir documentos que podem atingir dimensões consideráveis (Andrade, 2015).

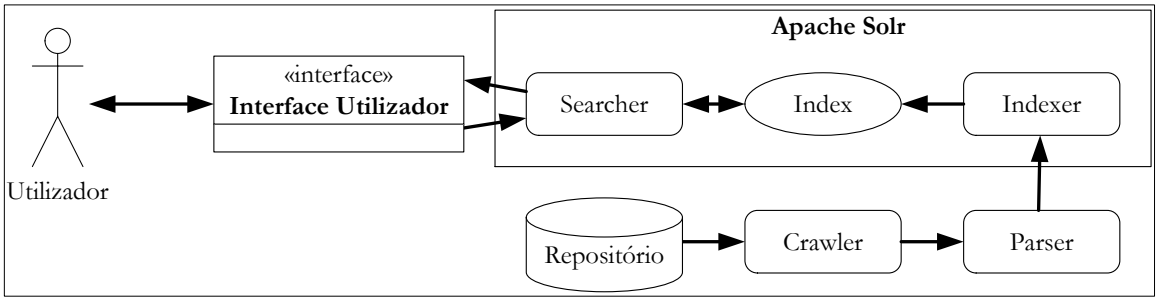


Figura 19 - Componentes de um sistema empresarial de pesquisa com integração da plataforma Apache Solr

5.2 Mecanismo de procura - crawler

Esta é a primeira etapa de desenvolvimento e implementação do protótipo associado ao sistema CYCLOPS e tem por objetivo percorrer os repositórios (*SVN e AsknowledgeDb)* para recolher documentos de acordo com a consulta feita pelo utilizador. Para este fim foi desenvolvido um *crawler* específico para cada um dos repositórios, conforme se apresenta de seguida, Figura 20.

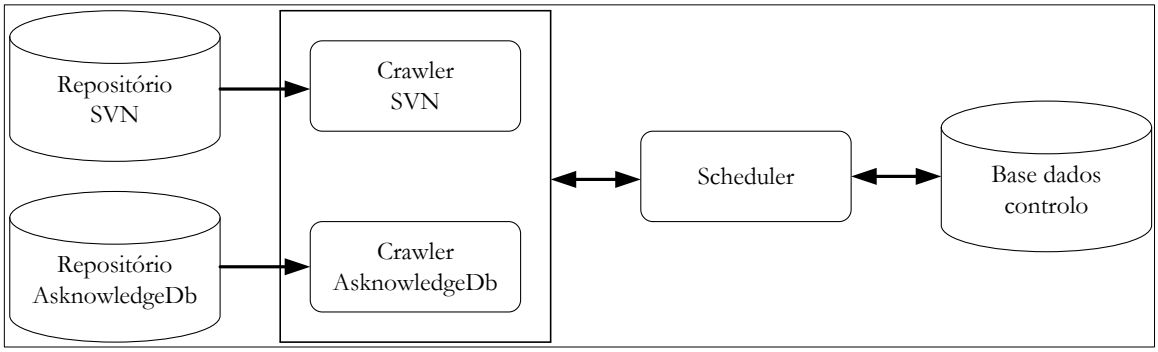


Figura 20 - Bloco responsável pela procura e recolha de dados dos repositórios *SVN* e *AcknowledgeDb*

Para proceder à procura de documentos em cada um repositórios tratados no âmbito do desenvolvimento e implementação deste protótipo (repositório *subversion* (*SVN*) e repositório *AsknowledgeDb*), foram desenvolvidos dois *crawlers*, cada um deles tendo em atenção a especificidade de cada um dos repositórios, designadamente:

- *crawler* associado ao repositório *SVN* - este *crawler* percorrerá, genericamente, todo o repositório verificando o número da sua versão (este número corresponde às alterações de estado do repositório). Se este número corresponde ao número da versão guardada na base de dados de controlo o *crawler* nada tem que fazer (não houve

mudança de estado do repositório). Se os números comparados forem diferentes o *crawler* verifica que alterações ocorreram entre revisões e faz *publish* destas alterações para o sistema de filas onde serão posteriormente transformados e indexados. Se o número da versão na base de dados de controlo for -1, o *crawler* percorrerá todo o repositório à procura de ficheiros válidos.

- *crawler* associado à base de dados *AsknowledgeDb* - neste caso o *crawler* verifica o número da versão da base de dados de controlo. Se este número for -1 o *crawler* percorrerá todo o repositório à procura de ficheiros válidos. Caso contrário o *crawler* procurará as últimas alterações no repositório. Em qualquer dos casos os documentos recolhidos pelo *crwaler* são enviados para um sistema de filas que os encaminhará para o *parser*.

O sistema de filas fará a ligação entre o mecanismo de procura, através de *Crawling* e o mecanismo de transformação, que através de um *parser* interpretará a estrutura do documento, ver Figura 21. Para assegurar a gestão destas filas é utilizada a plataforma Apache Kafka.

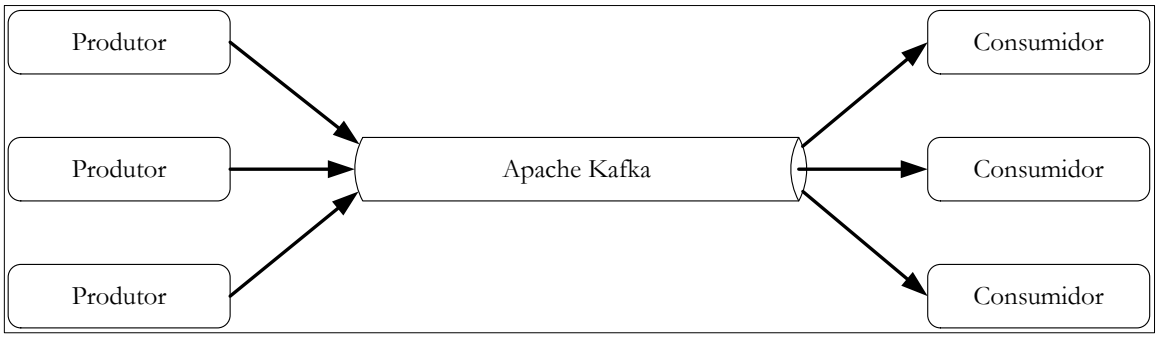


Figura 21 - Apache Kafka como plataforma gestora de mensagens entre produtor e consumidor

Conforme já referido, os *crawlers* associados ao protótipo do sistema CYCLOPS foram desenvolvidos usando a biblioteca padrão Java 8 e têm capacidade para recolher qualquer tipo de ficheiro existente no repositório *SVN*, considerando os mais diversos tipos de formato (p. e. Microsoft Word (.doc); Microsoft PowerPoint (.ppt); Microsoft Excel (.xlsx) e documentos pdf (.pdf); etc.) e também no repositório *AsknowledgeDb*.

5.3 Mecanismo de transformação - parser

Nesta etapa, os documentos recolhidos pelo *crawler* na fase de pesquisa e recolha de documentos nos repositórios (*SVN e AsknowledgeDb*) serão analisados e tratados em função do tipo de dados. Para este efeito foi igualmente desenvolvido um *parser*, ver Figura 22.

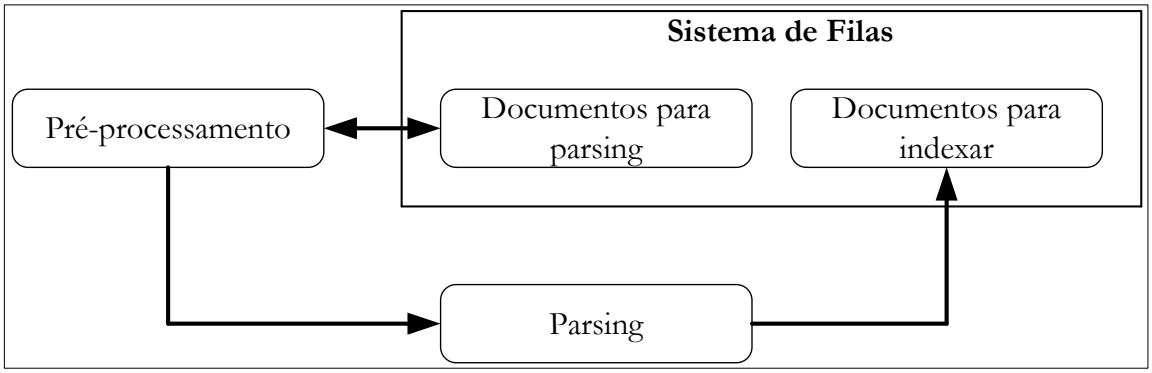


Figura 22 - Bloco responsável pela transformação dos documentos recolhidos no repositório *SVN*

Após recolher os documentos é necessário proceder ao reconhecimento do seu conteúdo e da sua estrutura, através de um *parser*. No desenvolvimento e implementação deste protótipo vamo-nos focar na análise dos metadados, informações sobre um documento que não fazem parte do conteúdo do texto. Assim, o *parser* recorre a *tags* e outros metadados reconhecidos no documento para interpretar a sua estrutura com base na sintaxe da linguagem de marcação (análise sintática) e para produzir uma representação do documento que inclui tanto a estrutura quanto o conteúdo (Croft et al., 2015).

Nesta etapa, como o nome indica, os ficheiros encontrados pelo *crawler* são transformados para um formato que facilite a sua indexação e são retiradas informações que caracterizam o documento (metadados) como por exemplo: título, corpo do texto, conteúdo, autor. Assim, o *parser* tem como função, com base no tipo de documento que este trata, retirar os metadados do documento, identificar o seu metatipo e proceder ao seu envio para um *publisher* que terá como função enviar o documento para uma *queue*. Aqui devem ser considerados três importantes componentes, a saber: (1) *subscriber*, (2) *parser* e (3) *publisher*.

A extração de conteúdo e metadados dos documentos recebidos pelo *parser* é feita com recurso ao Apache Tika.

5.4 Mecanismo de gestão de mensagens – Apache Kafka

O fluxo de eventos entre os vários blocos que constituem o protótipo do sistema CYCLOPS é assegurado por um sistema de filas. Para implementar este sistema foi utilizada a plataforma Apache Kafka, ver Figura 23.

O sistema de filas entre o *crawler* e o *parser* guarda os documentos recolhidos pelo *crawler* que com base no tipo de repositório procede ao seu envio para o *parser* respetivo ou que com base na operação (se for de eliminação) procede ao seu envio para o *indexer*. Este sistema de filas pode ser também utilizado para guardar documentos provenientes do *parser* e encaminhados para o *indexer*.

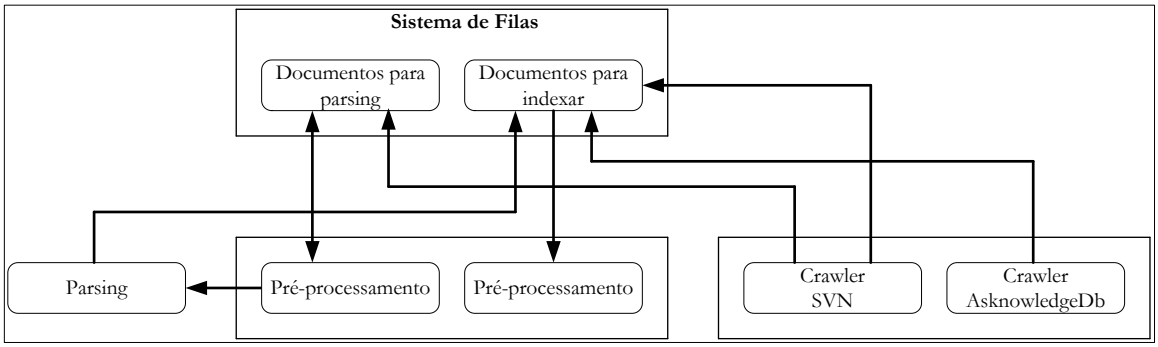


Figura 23 - Bloco responsável pelo sistema de gestão de mensagens

A nossa opção pelo sistema de mensagens distribuído Apache Kafka fica a dever-se a algumas características oferecidas por esta tecnologia no âmbito do desenvolvimento do presente projeto: velocidade superior à que é apresentada pelas tecnologias concorrentes; separação das mensagens por tópicos e fácil integração com outras tecnologias Apache (Taveira, 2015) assim como a capacidade de persistência até a mensagem ser processada.

A plataforma Apache Kafka é um mecanismo de troca de mensagens em que o responsável pela produção da mensagem não tem necessidade de conhecer quem vai consumir essa mensagem. Assim, as mensagens são organizadas por tópicos e podem ser lidas sem ser necessário haver conhecimento de quem as produziu, ver Figura 24.

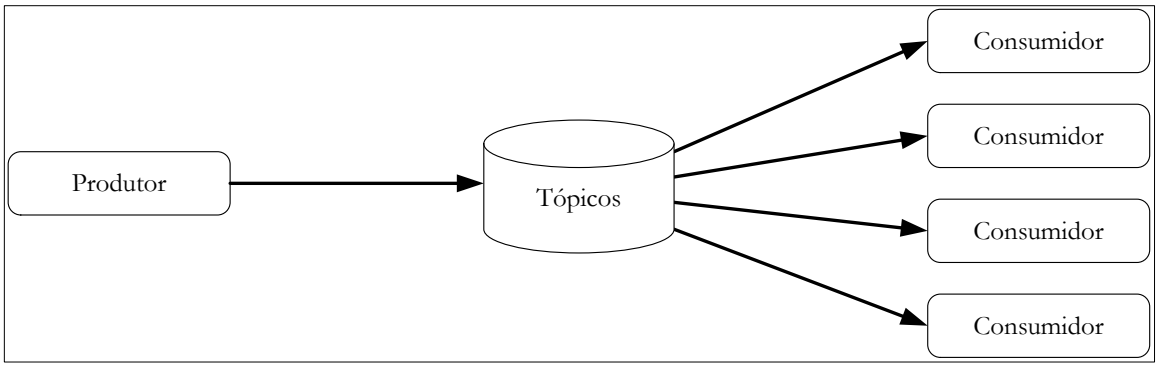


Figura 24 - Mecanismo de troca de mensagens organizadas por tópicos

5.5 Mecanismo de indexação - indexer

Este mecanismo, no essencial, é da competência da plataforma de pesquisa de informação empresarial Apache Solr. É usada uma biblioteca externa *solrj* de modo a integrar a plataforma Apache Solr com a linguagem de programação Java, facilitando assim a realização das operações de adição, modificação e atualização com recurso à mesma linguagem, ver Figura 25.

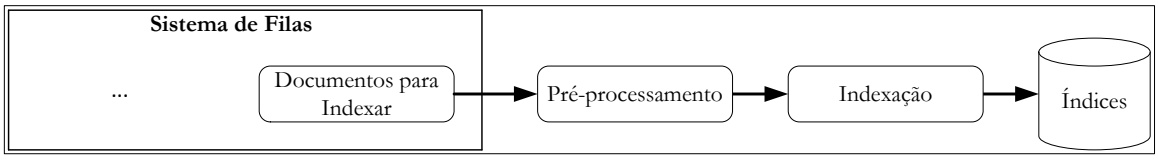


Figura 25 - Bloco responsável pelo processo de indexação

O Apache Solr é um servidor de pesquisa empresarial, baseado na plataforma de pesquisa Apache Lucene. O Apache Solr disponibiliza algumas características relevantes, nomeadamente o processamento e indexação de vários tipos documentos, além de ser uma plataforma escalável.

O Apache Lucene apenas contém o núcleo do sistema de pesquisa, não tendo incorporado qualquer *crawler* ou *parser*, devendo estas funcionalidades ser adicionadas, pelo desenvolvedor, ao protótipo desenvolvido e implementado no decorrer do estágio a que respeita o presente relatório.

O Apache Solr dispõe de algoritmos específicos para calcular um *ranking*, baseado na frequência e duração do termo no documento, associado aos documentos que permite determinar a sua relevância e portanto a conformidade entre os resultados obtidos e o termo utilizado para efetuar a pesquisa.

Relativamente à relevância pode verificar-se que, quando um documento é indexado no Apache Solr, as várias informações que lhe são referentes são indexados em vários campos daquela plataforma, estes campos são, designadamente: o título, o conteúdo, o autor, a data e a identificação do repositório onde o documento se encontra armazenado. Assim, quando se realiza uma pesquisa de um documento através da interface *Web*, é feita uma procura, tomando como referência as *keywords* digitadas na caixa de pesquisa, em todos aqueles campos, provocando um "*boost*" e o consequente aumento da relevância de certos campos (p. ex. título, autor, etc.). Assim, se as *keywords* se encontrarem nestes campos o documento procurado será considerado mais relevante.

5.6 Lista de controlo de acesso

Considerando que cada repositório permite acesso apenas a utilizadores autorizados, quando se indexam ficheiros em repositórios de qualquer tipo deve garantir-se que apenas utilizadores autorizados podem utilizar a interface *Web* para procurar documentos e obter documentos a que têm permissão de acesso. Assim cada *crawler* é responsável por enviar o ficheiro assim como a lista dos utilizadores que têm permissão para lhe aceder. No caso do *crawler* do repositório *SVN*, se o utilizador for considerado administrador, estando o seu *username* no grupo admin para esse repositório, tem acesso ao ficheiro de configuração do repositório. Com este ficheiro é possível verificar que utilizadores existem em cada grupo, e a que pastas cada grupo tem acesso, com esta informação, quando o *crawler* encontra um ficheiro considerado válido para indexação, este verifica que utilizadores têm acesso ao mesmo e envia esta informação para o *parser*.

No caso do *crawler AsknowledgeDb* este processo é mais simples uma vez que é considerado um repositório público para todos os colaboradores da empresa, neste caso é assumido que todos eles têm acesso aos ficheiros que residem no mesmo.

Estas permissões são indexadas também no Apache Solr, mas num ficheiro de índex diferente. Isto é feito para facilitar as operações de atualização uma vez que não é preciso proceder, novamente, à indexação do ficheiro, sendo apenas necessário modificar este ficheiro de índex.

Na altura de pesquisa todas as consultas realizam um *join* entre os ficheiros de índex, de modo a verificar se o utilizador tem permissões para visualizar os ficheiros.

5.7 Interface *Web*

A interface Web, ver Figura 26, tem por objetivo facilitar aos utilizadores a utilização do sistema CYCLOPS, de modo que possam realizar as suas pesquisas tendo em atenção os documentos que foram indexados pelo sistema. Este sistema disponibilizará um mecanismo de *login* que estará integrado com o LDAP institucional, de modo a permitir a autenticação de cada um dos utilizadores tendo presente o perfil que foi atribuído, previamente, pela empresa a cada um dos seus colaboradores, este perfil já é utilizado pelo repositório *SVN* para a ACL.

Existe um sistema de permissões hierárquico que, conforme o cargo de um utilizador da empresa, dar-lhe-á mais ou menos informação. Este sistema está organizado por níveis: o nível mais baixo apenas dá acesso à informação a que o utilizador tem de facto permissões; nos níveis intermédios, se o utilizador procurar informação à qual não tem acesso aparecerá indicação de que a informação existe mas o utilizador não têm permissões para lhe aceder. Por fim no nível mais alto o sistema mostra todos os documentos.

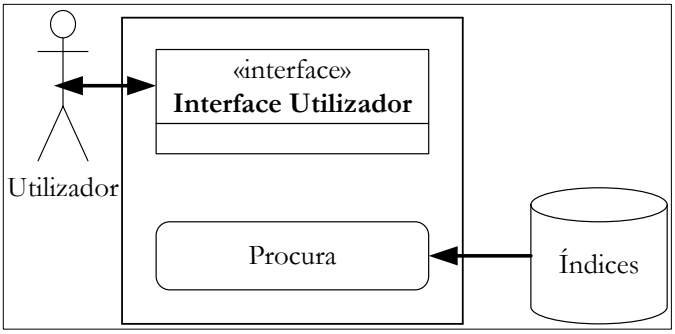


Figura 26 - Bloco responsável pela interface Web

A interface *Web*, desenvolvida, tem o aspeto que se pode ver na Figura 27.

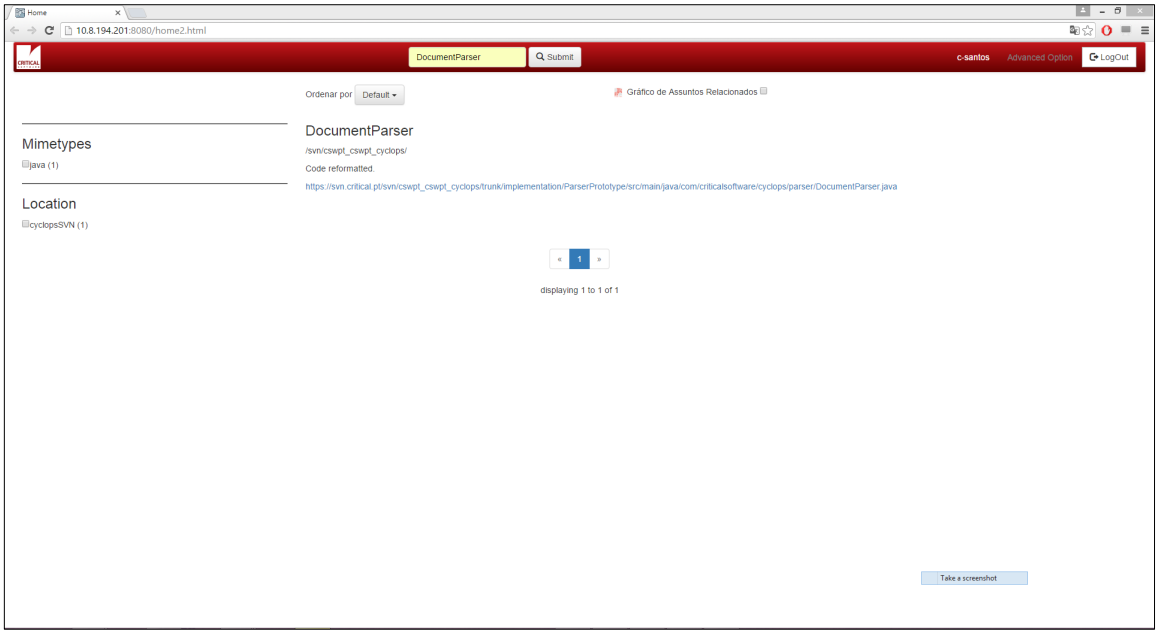


Figura 27 - Interface Web do protótipo do sistema CYCLOPS

[Esta página foi intencionalmente deixada em branco]

Capítulo 6
Testes e Verificação

Na fase de especificação do conjunto de testes devem ser identificadas, as partes a serem testadas e as condições de teste para cada uma dessas partes, como primeiro passo para definição de cada um dos casos de teste a serem executados (ISO/IEC/IEEE 29119-3, 2013).

Estes testes podem ser manuais e/ou automatizados, tendo ambos como objetivo melhorar a qualidade do sistema a ser testado através da sua verificação e validação (Bernardo, 2011). Os testes manuais têm intervenção humana, enquanto que os testes automatizados são totalmente automáticos procurando-se assim evitar o erro que existe sempre associado ao comportamento humano.

A importância dos testes automatizados tem vindo a ser reconhecida por várias ferramentas de software que implementam módulos para integração e realização de testes. A ferramenta Apache Maven é disto exemplo ao fornecer uma arquitetura padronizada para criação e execução de casos de teste automatizados (Bernardo, 2011).

De acordo com o IEEE (2014) o teste de software consiste na verificação dinâmica do comportamento esperado de um programa para um conjunto finito de casos de teste, adequadamente selecionados a partir do domínio de execução normalmente infinito.

Assim, face ao exposto anteriormente, para testar o protótipo do sistema CYCLOPS foi identificado um conjunto de funcionalidades a serem testadas, assim como foram identificadas outras funcionalidades que não carecem de ser testadas. Para cada uma das funcionalidades a serem testadas foram definidos os respetivos casos de teste, optando-se, sempre que possível, pela sua automatização.

6.1 Funcionalidades testadas

Para definir o plano de testes, para testar o protótipo do sistema CYCLOPS, subdividiu-se o sistema em dois grandes blocos: (1) sistema propriamente dito e (2) interface *Web*, descritos nas subsecções seguintes. Para realização dos testes ao sistema propriamente dito é feita a sua separação por componentes: (*crawler*, *parser*, *indexer*), os quais são testados individualmente, recorrendo a testes automatizados. Para realização dos testes da interface *Web* são testadas todas as suas funcionalidades, recorrendo, neste caso, a testes manuais.

Relativamente ao *crawler* serão testadas as seguintes funcionalidades, recorrendo a testes automatizados, implementados com recurso à *framework* JUnit que facilita a criação de código para automatização dos referidos testes:

- quantidade de documentos armazenados no repositório;
- tamanho dos documentos armazenados no repositório;

- integridade dos documentos armazenados no repositório;
- controlo de acesso aos documentos armazenados no repositório.

No que respeita ao *parser* serão testadas as seguintes funcionalidades, recorrendo igualmente à *framework* JUnit:

- extração do conteúdo do documento;
- documentos recebidos pelo *parser*.

Em relação ao *indexer* serão testadas as funcionalidades listadas de seguida, implementando igualmente código baseado na *framework* JUnit automatizando os respetivos casos de teste:

- documentos recebidos pelo *indexer*:
- documentos indexados.

No que respeita à interface *Web*, serão testadas as seguintes funcionalidades, recorrendo a testes manuais:

- login/logout de utilizador com conta da CRITICAL Software, SA;
- disposição dos elementos gráficos conforme especificação;
- assegurar que todas as funcionalidades que resultam de ações de utilizadores interagem como especificado;
- validar os resultados que são mostrados ao utilizador.

6.2 Funcionalidades não testadas

Fora do âmbito deste documento ficam os testes às plataformas utilizadas: Apache Solr, Apache Kafka e Apache Tika. Pois embora estas tecnologias sejam utilizadas pelo sistema, estão suficientemente testadas conforme podemos concluir através da revisão da literatura e do estado da arte. Os sistemas e serviços externos que interagem com o sistema CYCLOPS não são, igualmente, considerados neste plano de testes.

6.3 Abordagem

Relativamente às funcionalidades a serem testadas relativamente a cada um dos módulos (*crawler*, *parser*, *indexer* e interface *Web*) foram criados testes genéricos para cada um deles. Uma vez que, futuramente, poderão surgir novas variantes destes módulos deverá ser possível proceder ao seu teste de forma similar. Esta situação beneficiará da implementação, sempre que possível, de testes automatizados.

Para definição destes testes procedeu-se à análise dos requisitos e funcionamento de cada componente de modo a identificar características comuns em cada um dos módulos individualmente. Posteriormente foram definidos os passos que serão obrigatórios para testar cada módulo, podendo estes serem complementados de forma a poderem lidar com as especificidades de funcionamento de cada um dos módulos.

Embora os casos de teste apresentados tenham sido executados para os repositórios (*subversion SVN e AsknowledgeDb*), somente é apresentada a implementação e os resultados obtidos para o repositório *SVN*. Para o repositório *AsknowledgeDb* tudo acontece de forma similar à exceção da quantidade de documentos armazenados em cada um dos repositórios (244 no *SVN* e 3718 no *AsknowledgeDb*).

6.4 Crawler - casos de teste

Com base na identificação das necessidades a serem testadas foram organizados casos de teste para o *crawler*, conforme se apresenta seguidamente.

6.4.1 Quantidade de documentos armazenados no repositório

Este caso de teste testa o *crawler*, tendo em atenção a quantidade de documentos armazenados no repositório pesquisado, o objetivo deste teste é verificar se todos os documentos são analisados quando se executa uma pesquisa a um determinado repositório, ver Quadro 1.

Para executar o *crawler* com base neste caso de teste recorreu-se, conforme já referido, à *framework* JUnit, facilitadora de testes automatizados. De acordo com Bernardo (2011) esta *framework* foi desenvolvida em 1998 por Kent Beck e Enrich Gamma, tendo tido um papel relevante como facilitadora do desenvolvimento da prática de testes automatizados.

```
@Test
public void numberFiles(){
  int sum = crawler.aftercount;
  assertEquals(244,sum);
}
```

Quadro 1 - Código que implementa o caso de teste "quantidade de documentos"

Conforme podemos verificar pelos resultados devolvidos por este caso de teste, Tabela 33, a sua execução foi feita com sucesso, não tendo sido identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Documentos não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 33 - Resultados do caso de teste "quantidade de documentos"

6.4.2 Tamanho dos documentos armazenados no repositório

Este caso de teste verifica o funcionamento do *crawler*, tendo em atenção o tamanho dos documentos armazenados nos repositórios pesquisados, o objetivo é testar se todos os documentos são analisados, independentemente do seu tamanho, ver Quadro 2. Este caso de teste foi igualmente executado com recurso à *framework* JUnit.

```
@Test
public void sizeFiles(){
  for(int i=0; i<244;i++){
    for(int j=0; j<244;j++){
      if(crawler.afterCrawl.get(j).getPath().equals(crawler.beforeCrawl.get(i).getPath())){
        assertEquals(crawler.beforeCrawl.get(i).getFileSize(),crawler.afterCrawl.get(j).getFileSize());
      }
    }
  }
}
```

Quadro 2 - Código que implementa o caso de teste "tamanho dos documentos"

Conforme podemos verificar pelos resultados devolvidos por este caso de teste, Tabela 34, a sua execução também foi feita com sucesso, não tendo sido identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Documentos não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 34 - Resultados do caso de teste "tamanho dos documentos"

6.4.3 Integridade dos documentos armazenados no repositório

Este caso de teste verifica o funcionamento do *crawler*, tendo em atenção a integridade dos documentos armazenados no repositório pesquisado após terem sido procurados pelo *crawler*, o objetivo é testar se todos os documentos mantêm a sua integridade após ser executada uma pesquisa aos documentos armazenados num determinado repositório, ver Quadro 3. Este caso de teste foi realizado, igualmente, recorrendo à *framework* JUnit.

```
@Test
public void checkFileContent(){
   for(int i=0; i<244;i++){
      for(int j=0; j<244;j++){
         if(crawler.afterCrawl.get(j).getPath().equals(crawler.beforeCrawl.get(i).getPath())){
            assertTrue(Arrays.equals(crawler.beforeCrawl.get(i).getFile(),crawler.afterCrawl.get(j).getFile()));
         }
      }
   }
}
```

Quadro 3 - Código que implementa o caso de teste "integridade dos documentos"

A execução deste caso de teste devolveu os resultados apresentados na Tabela 35, conforme se pode verificar, também aqui, não foi identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Ficheiros não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 35 - Resultados do caso de teste "integridade dos documentos"

6.4.4 Controlo de acesso aos documentos armazenados no repositório

Este caso de teste verifica o funcionamento do *crawler*, tendo em atenção o controlo de acesso aos documentos armazenados no repositório. O objetivo consiste em verificar se a ACL do repositório é corretamente associada a cada um dos documentos procurados, ver Quadro 4. Este caso de teste foi, igualmente, realizado com recurso à *framework* JUnit.

```
@Test
public void getDocumentACLs(){
List<String> perms  = Arrays.asList("naalves,hlsousa,c-santos");
for(int i=0;i<244;i++){
if(crawler.beforeCrawl.get(i).getPath().equals("C:\\var\\FileTemp\\admin\\AdminPermsTest.txt")){
         assertEquals(perms,crawler.beforeCrawl.get(i).getUsers());
      }
   }
}
```

Quadro 4 - Código que implementa o caso de teste "controlo de acesso aos documentos"

À semelhança do que aconteceu na execução dos casos de teste anteriores, este teste devolveu os resultados apresentados na Tabela 36 que, conforme se pode verificar garantem que a sua execução foi feita com sucesso, não tendo sido identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Ficheiros não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 36 - Resultados do caso de teste "controlo de acesso aos documentos"

6.5 Parser - casos de teste

6.5.1 Extração de conteúdo

Este caso de teste verifica o funcionamento do *parser*, tendo em atenção o conteúdo dos documentos enviados do *crawler* para o *parser*. O objetivo deste teste é verificar se o conteúdo extraído pelo *parser* após as transformações corresponde ao que existia no documento original, ver Quadro 5. Também, neste caso de teste, se recorreu à *framework* JUnit.

```
@Tes
public void fileContents(){
parser.run();
   for(int i = 0; i<244;i++){
      assertEquals(parser.beforeParse.get(i).getContent(),parser.afterParse.get(i).getContent());
   }
}
```

Quadro 5 - Código que implementa o caso de teste "extração de conteúdo dos documentos"

A execução deste caso de teste devolveu os resultados apresentados na Tabela 37, conforme se pode verificar, também aqui, não foi identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Ficheiros não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 37 - Resultados do caso de teste "extração de conteúdo dos documentos"

6.5.2 Documentos recebidos pelo parser

Este caso de teste verifica o funcionamento do *parser*, tendo em atenção os documentos enviados do *crawler* para o *parser*. O objetivo deste teste é verificar se todos os documentos recolhidos pelo *crawler* são enviados para o *parser*, ver Quadro 6. Caso de teste implementado com recurso à *framework* JUnit.

```
@Test
public void receivedFiles(){
parser.run();
   assertEquals(244,parser.documentCount);
}
```

Quadro 6 - Código que implementa o caso de teste "documentos recebidos pelo *parser*"

Os resultados devolvidos após execução deste caso de teste são mostrados na Tabela 38, conforme podemos verificar, também aqui não foi identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Ficheiros não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 38 - Resultados do caso de teste "documentos recebidos pelo *parser*"

6.6 Indexer - casos de teste

6.6.1 Documentos recebidos pelo indexer

Este caso de teste tem por objetivo verificar se o *indexer* recebe todos os documentos enviados pelo *parser*, ver Quadro 7. Caso de teste igualmente implementado com recurso à *framework* JUnit.

```
@Test
public void numberOfDocsEntry(){
   indexer.readDocs();
   assertEquals(244,indexer.documentReceivedCount);
}
```

Quadro 7 - Código que implementa o caso de teste "documentos recebidos pelo *indexer*"

Os resultados devolvidos após execução deste caso de teste são mostrados na Tabela 39, conforme podemos verificar, também aqui não foi identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Ficheiros não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 39 - Resultados do caso de teste "documentos recebidos pelo *indexer*"

6.6.2 Documentos indexados

Este caso de teste verifica o funcionamento do *indexer*, tendo em atenção os documentos enviados do *parser* para o *indexer*. O objetivo deste teste é verificar se o *indexer* faz a correta indexação dos documentos e se estes podem ser, mais tarde, procurados com recurso a uma interface *Web*, ver Quadro 8. Também, este caso de teste, foi implementado com recurso à *framework* JUnit.

```
@Test
public void numberOfIndexedDocs(){
assertEquals(244,indexer.numberOfDocumentsIndexed());
}
```

Quadro 8 - Código que implementa o caso de teste "documentos indexados"

Os resultados devolvidos após execução deste caso de teste são mostrados na Tabela 40, conforme podemos verificar, também aqui não foi identificada qualquer anomalia.

Documentos no repositório	Erros encontrados	Falhas no teste	Ficheiros não testados	Taxa de sucesso
244	0	0	0	100%

Tabela 40 - Resultados do caso de teste "documentos indexados"

6.7 Interface Web - casos de teste

Os testes à interface *Web* foram realizados manualmente, tendo sido executados seis testes conforme descrevemos de seguida.

6.7.1 Caixa de pesquisa

Este teste tem por objetivo verificar se após inserir um termo na caixa de pesquisa, seguindo o protocolo indicado abaixo, os resultados devolvidos são os esperados:

- passo 1 - inserir termo de pesquisa na caixa *search*;
- passo 2 – premir o botão *submit*.

Os resultados obtidos com a execução deste teste são os esperados, assim:

- são mostrados na parte direita do ecrã, até 10 resultados;
- é mostrado o menu de paginação no rodapé da página;
- o menu de filtros é mostrado na parte esquerda do ecrã (*sidebar*);
- cada um dos resultados mostrados mostra o *MIME type* do documento, o título, a localização (nome do repositório) e a descrição;
- o menu de filtros mostra os seguintes filtros: *MIME type*, autor, localização.

6.7.2 Menu sidebar

O objetivo deste teste consiste em verificar o funcionamento do menu *sidebar*, sendo executado de acordo com o seguinte protocolo:

- passo 1 - inserir termo de pesquisa na caixa *search;*
- passo 2 - premir o botão *submit;*
- passo 3 - premir qualquer um dos botões no menu esquerdo;
- passo 4 - verificar os resultados filtrados mostrados no menu à direita.

Os resultados obtidos com a execução deste segundo teste, são os seguintes:

- são mostrados na parte direita do ecrã, até 10 resultados;
- é mostrado o menu de paginação no rodapé da página;
- o menu de filtros é mostrado na parte esquerda do ecrã (*sidebar*);
- cada um dos resultados mostrados mostra o *MIME type* do documento, o título, a localização (nome do repositório) e a descrição;
- o menu de filtros mostra os seguintes filtros: *MIME type*, autor, localização;
- cada um dos filtros tem um número de botões equivalente ao número de elementos únicos da pesquisa;
- no caso do *MIME type* aparece um número de botões igual ao *MIME type* existente assim como o número de documentos por *MIME type*;
- no caso do autor é mostrado um número de botões igual ao número de autores existente assim como o número de documentos por autor;
- no caso do autor é mostrado um número de botões igual ao número de localizações (repositórios) existente assim como o número de documentos por repositório;
- após premir num qualquer um dos botões os resultados que são mostrados à direita são filtrados com base nos botões selecionados.

6.7.3 Paginação

O objetivo deste terceiro teste consiste em verificar o funcionamento do menu paginação, sendo utilizado o seguinte protocolo:

- passo 1 - inserir o termo de pesquisa na caixa que *search*;
- passo 2 – premir o botão *submit*.

Os resultados obtidos com a execução deste teste, são os seguintes:

- são mostrados na parte direita do ecrã, até 10 resultados;
- caso o número de resultados seja superior a 10, deverão ser apresentados os respetivos botões no menu de paginação;
- é mostrado um botão por cada 10 resultados, sendo numerados de 1 até n;
- premindo o botão de paginação devem ser mostrados até 10 resultados, ordenados pelo motor de pesquisa. Exemplo: ao premir o botão 1 são mostrados os primeiros 10 resultados devolvidos pelo Apache Solr, ordenados pela sua relevância.

6.7.4 Logout

Consiste este teste em verificar o funcionamento do botão *logout*, sendo utilizado o seguinte protocolo:

- passo 1 - realizar *login* na aplicação CYCLOPS;
- passo 2 – premir o botão *lougout*.

Com a execução deste teste obtivemos os seguintes resultados:

- ao premir o botão *logout*, o utilizador é encaminhado para a página de *login*, não sendo autorizado a efetuar nova pesquisa sem realizar, novamente, o *login*.

6.7.5 Login

Consiste este teste em verificar o funcionamento do botão *login*, sendo utilizado o seguinte protocolo:

- passo 1 – inserir o *username* e a *password* do utilizador.

Com a execução deste teste obtivemos os seguintes resultados:

- após inserir os dados de *login* corretos o utilizador é encaminhado para a página inicial de pesquisa;
- caso os dados de *login* sejam inseridos de forma incorreta o utilizador permanecerá na mesma página sendo informados de que os dados foram introduzidos de forma incorreta.

6.7.6 Permissão/autorização

Finalmente o objetivo deste teste consiste em verificar se o sistema tem mecanismos que permitam identificar corretamente um qualquer utilizador, devidamente autorizado, o protocolo utilizado é o seguinte:

- passo 1 – o utilizador efetua o *login* na aplicação CYCLOPS.

Os resultados obtidos com a execução deste teste são os seguintes:

- ao entrar na página inicial de pesquisa, o nome do utilizador (*login username*), é mostrado no canto superior direito.

6.8 Teste de carga à interface Web

Foram feitos testes de carga à interface *Web* utilizando a ferramenta Apache JMeter, utilizada para execução dos mesmos. Os resultados mostrados na Figura 28 resultam da execução de um teste que simula a utilização do sistema por 25 utilizadores.

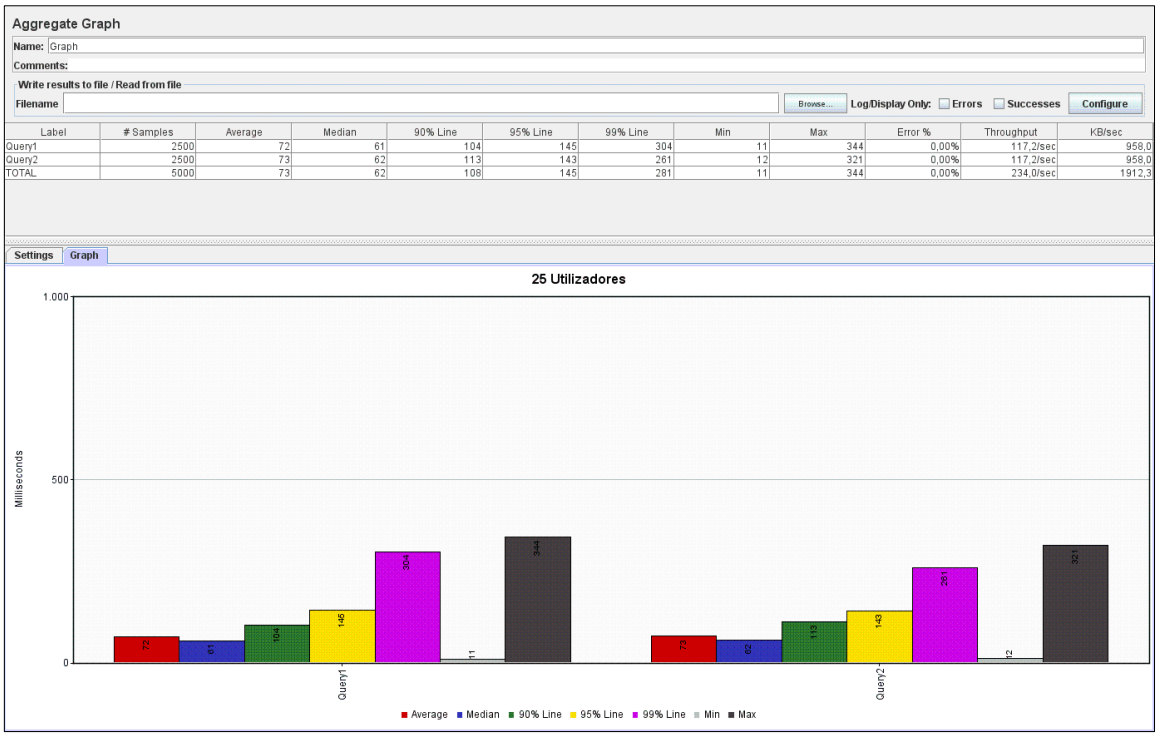


Figura 28 - Resultados do teste de carga para 25 utilizadores

Os resultados mostrados na Figura 29 resultam da execução de um teste que simula a utilização do sistema por 50 utilizadores.

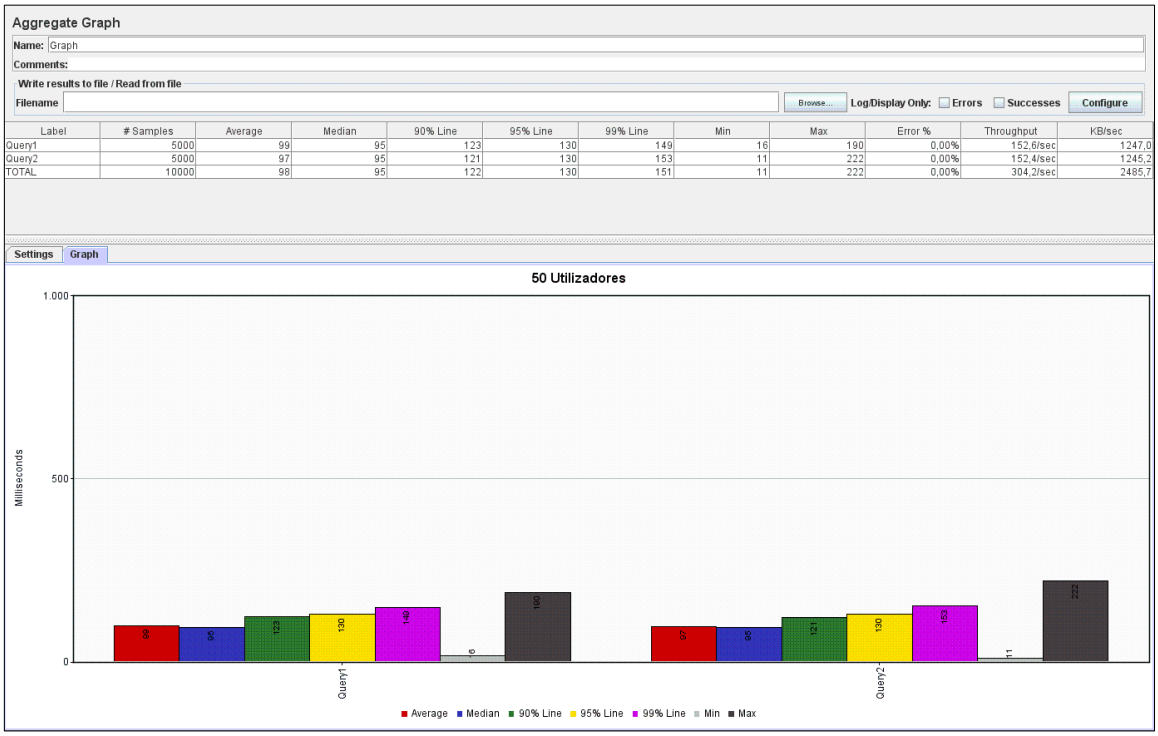


Figura 29 - Resultados do teste de carga para 50 utilizadores

Os resultados mostrados na Figura 30 resultam da execução de um teste que simula a utilização do sistema por 75 utilizadores.

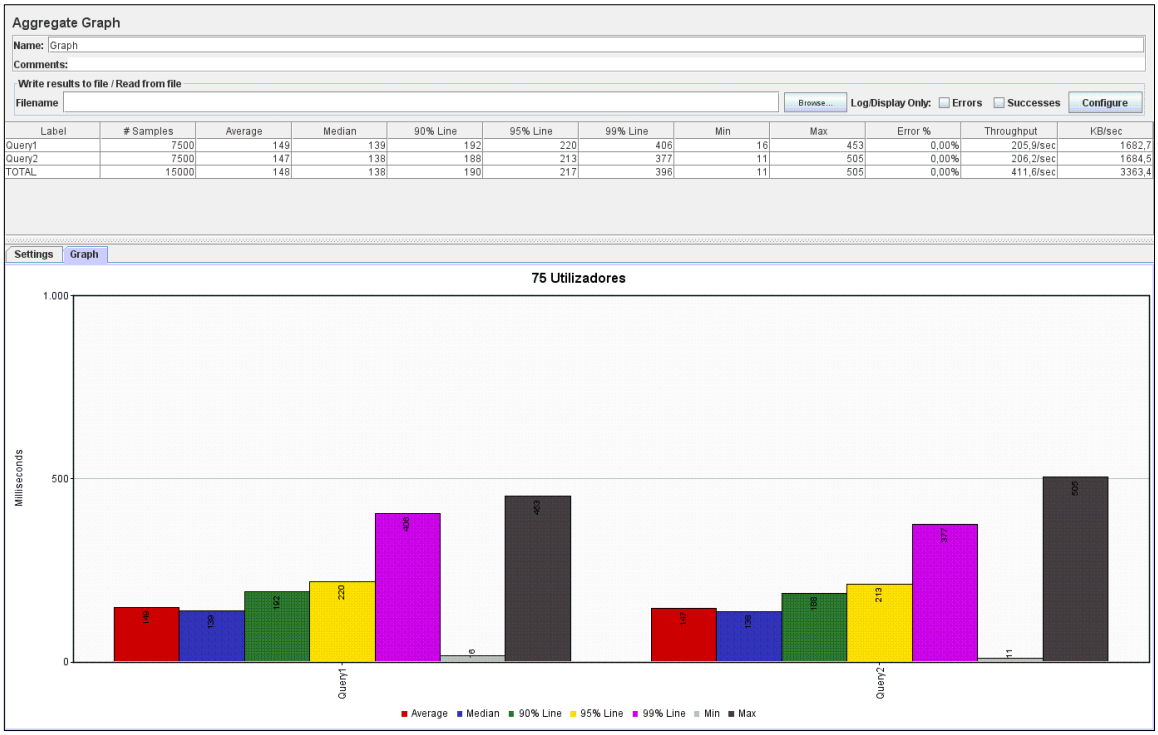


Figura 30 - Resultados do teste de carga para 75 utilizadores

Os resultados mostrados na Figura 31 resultam da execução de um teste que simula a utilização do sistema por 100 utilizadores.

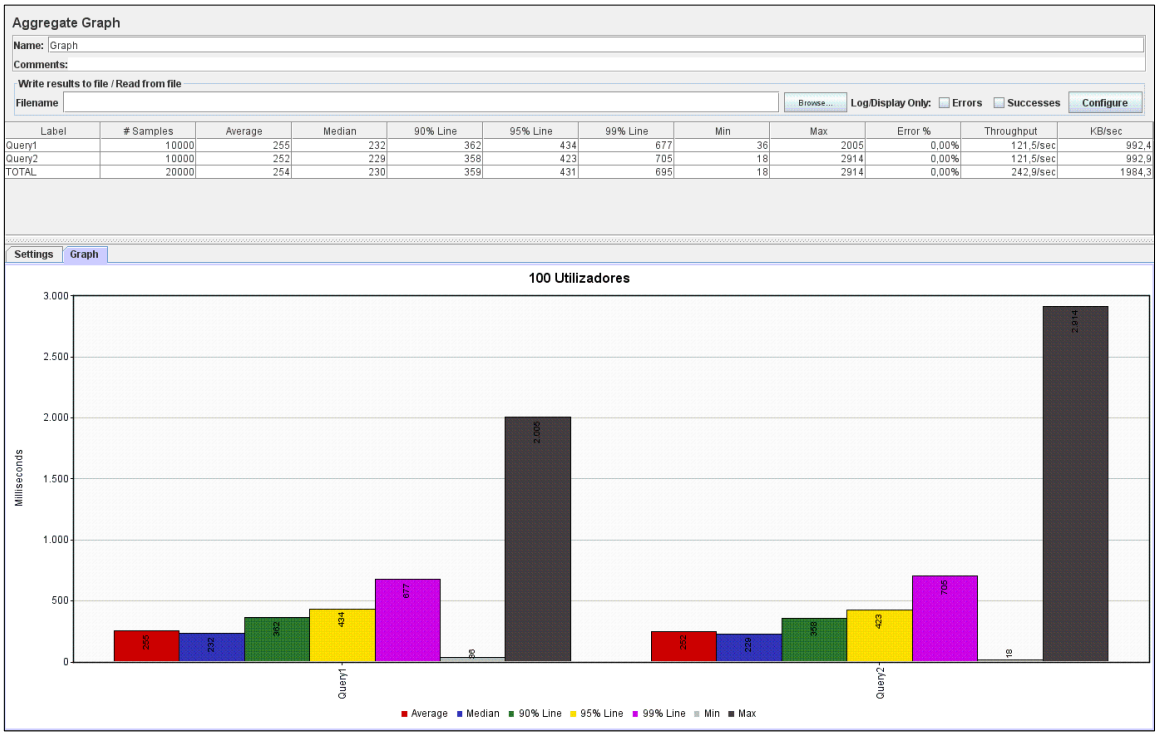


Figura 31 - Resultados do teste de carga para 100 utilizadores

Estes testes foram executados considerando que os utilizadores estão ligados em simultâneos e que cada pedido (Query1 e Query2) foi executado 100 vezes. Os dois pedidos correspondem a pedido HTTP GET, tendo cada um deles como argumento as palavras-chave introduzidas na caixa de pesquisa da interface *Web*.

[Esta página foi intencionalmente deixada em branco]

Capítulo 7
Conclusões e propostas de trabalho futuro

Neste capítulo apresentamos, para além das conclusões, as dificuldades encontradas e sugestões para trabalho futuro que possa, de alguma forma complementar aquele que aqui foi desenvolvido.

No desenvolvimento deste projeto o estagiário necessitou de se adaptar a uma forma de trabalhar muito próxima daquilo que encontrará na realidade. Esta situação, fez com que o estagiário, obviamente, adquirisse novas competências e se habituasse a novos hábitos de trabalho.

O protótipo desenvolvido neste projeto de estágio vem reforçar aquilo que já alguns trabalhos relacionados referem como relevante que é a utilização de tecnologias de código aberto como forma de minimizar os custos de desenvolvimento e manutenção de sistemas empresariais de pesquisa.

Para além da utilização destas tecnologias de código aberto foi necessário desenvolver alguns módulos, designadamente os *crawlers* (um por cada um dos repositórios considerados neste projeto), o *parser*, o indexador e a *interface Web* e os sistemas de filas, responsáveis pela gestão de mensagens entre componentes do sistema. O desenvolvimento dos *crawlers*, *parser* e do indexador foi feito com recurso à linguagem Java 8, a *interface Web* foi desenvolvida com recurso à linguagem HTML5, CSS e *Javascript* e os sistemas de filas foram implementados com recurso ao Apache Kafka. Em nossa opinião o desenvolvimento destes módulos pode trazer contributos importantes para esta área de conhecimento.

Deve ser salientado que para além da procura de informação em dois repositórios específicos (*subversion SVN e AsknowledgeDb*) a extensão da segunda fase do projeto, abrangendo os meses de julho e agosto, permitiu que fosse ainda desenvolvido um mecanismo de procura para um repositório *file system*.

Terminado o projeto de estágio, pode concluir-se que os objetivos apresentados no Capítulo 3 foram atingidos. Atingir estes objetivos era uma meta essencial para se poder dar por cumprido o desafio colocado pelo presente projeto de estágio que consistiu em desenvolver e implementar um protótipo do sistema CYCLOPS.

Os resultados dos testes realizados são bastante animadores, quanto à qualidade do protótipo desenvolvido, o que permite concluir que a arquitetura utilizada para desenvolvimento do protótipo é adequada garantindo os níveis de qualidade pretendidos para este projeto.

A opção pela utilização de tecnologias *open source*, designadamente as tecnologias mantidas pela Apache software Foundation, face aos resultados verificados parece ter sido uma opção correta, facilitando o desenvolvimento do projeto e minimizando os seus custos de desenvolvimento e manutenção.

O protótipo desenvolvido e implementado é a prova de conceito que confirma a exequibilidade do desenvolvimento e implementação do sistema CYCLOPS que está na base da realização do presente estágio curricular.

Na primeira fase deste estágio que decorreu entre setembro de 2015 e janeiro de 2016, foram encontradas dificuldades relacionadas com a imersão num ambiente real, bastante diferente do ambiente académico.

No entanto, estas dificuldades foram ultrapassadas e na segunda fase do estágio, que decorreu entre fevereiro e junho de 2016 (foi decidido introduzir algumas melhorias no trabalho desenvolvido, daí esta segunda fase prolongar-se por julho e agosto), o trabalho desenvolvido decorreu em conformidade com o planeamento feito para esta fase.

Como trabalho futuro poderão ser realizadas vários desenvolvimentos, nomeadamente: paralelismo nos *crawlers*, uma vez que é usado o Apache Zookeeper, pois o Apache Kafka o exige; poderá igualmente ser desenvolvida uma solução em que os *crawlers* dividam o trabalho a realizar e guardem o estado de cada *thread* no Apache Zookeeper.

Ainda como desenvolvimento futuro poderá ser desenvolvida uma solução com algumas vantagens, utilizando o Solr Cell, uma vez que proporciona *sharding* aos *indexes* do Apache Solr. Esta característica poderá ser muito útil caso se pretenda aumentar a performance do sistema empresarial de pesquisa face ao crescimento dos *indexes*.

Um outro desenvolvimento futuro deste protótipo poderá ter como objetivo extrair certas partes do documento de forma a conseguir melhorar os resultados da procura, por exemplo extrair a *table of contents* dos documentos word, e adicionar um novo campo ao Apache Solr. Após este processo é realizado um "*boost*" ao novo campo aumentando assim a relevância dos documentos se as keywords aparecerem neste campo. Este desenvolvimento poderá ser feito recorrendo ao Apache Poi (*framework* para lidar com documentos word e pdf entre outros).

O presente protótipo pode ainda deve ser desenvolvido para abranger qualquer dos repositórios corporativos existentes na CRITICAL Software, SA. Assim, é sugerido que sejam desenvolvidos os adequados *crawlers* tendo em atenção a especificidade desses mesmos repositórios.

Tendo em atenção o referido no Capítulo 2 pode concluir-se que se avizinha uma fase de mudança para o paradigma *bigdata* em que o volume de informação digital, criada e replicada, em todo o mundo cresce de forma exponencial. Assim é sugerido, igualmente como trabalho futuro, que o sistema desenvolvido seja adaptado para funcionamento neste tipo de ambiente.

Há outras tecnologias de código aberto, mantidas pela Apache Software Foundation ou não que poderiam ter sido utilizadas, tendo a nossa escolha recaído naquelas que são identificadas neste relatório. No entanto pensamos que será muito útil para desenvolvimento de sistema empresariais de pesquisa que outros estudos sejam desenvolvidos utilizando essas tecnologias alternativas e fazendo as devidas comparações.

Neste projeto, por razões de tempo, não foi possível testar a escalabilidade do sistema. Assim, seria bastante importante que estudos similares fizessem estes testes e comparassem diferentes tecnologias de forma a concluir sobre aquelas que melhor garantem a escalabilidade de um sistema empresarial de pesquisa.

Referências

Alberts, C. J., & Dorofee, A. J. (2010). *Risk Management Framework* (Technical report No. CMU/SEI-2010-TR-017; ESC-TR-2010-017). Software Engineering Institute, Carnegie Mellon University.

Alhabashneh, O., Iqbal, R., Shah, N., Amin, S., & James, A. (2011). Towards the Development of an Integrated Framework for Enhancing Enterprise Search Using Latent Semantic Indexing. In *Conceptual Structures for Discovering Knowledge* (Vol. 6828, pp. 346–352). Springer.

Almeida, B. (2014). *Dematerialization of Information Management Processes* (Dissertação de Mestrado). Universidade de Lisboa, Faculdade de Ciências,Departamento de Informática, Lisboa, Portugal.

Alter, S. (2013). Work system theory: Overview of core concepts, extensions, and challenges for the future. *Journal of the Association for Information Systems*, *14*(2), 72–121.

Alves, L. (2010). *Search Technologies, the Future Enterprise Applications* (Mestrado). Faculdade de Engenharia, Universidade do Porto, Porto.

Andrade, B. (2015). *Gestão e análise de eventos de segurança: uma abordagem ao transporte e armazenamento de eventos em ambientes Big Data* (Mestrado em Engenharia Informática). Escola de Engenharia, Departamento de Informática, Universidade do Minho, Braga.

Andrews, W., & Koehler-Kruener, H. (2015). *Magic Quadrant for Enterprise Search* (No. ID:G00269182). Gartner.

Arnold, S. (2014). *Redefining Search: Enterprise Search and Big Data*.

Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search* (Second edition.). Addison-Wesley.

Barbacci, M., Klein, M. H., longstaff, T. A., & Weinstock, C. B. (1995). *Quality attributes* (Technical Report No. CMU/SEI-95-TR-021 - ESC-TR-95-021). Pittsburgh, Pennsyilvania: Software Engineering Institute, Carnegie Mellon University.

Barbosa, A., Azevedo, B., Pereira, B., Campos, P., & Santos, P. (2007). Metodologia Ágil: Feature Driven Development.

Barrigas, H. (2014). *WISE – Web Information System Enterprise* (Dissertação de Mestrado). Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, Portugal.

Benghozi, P.-J., & Chamaret, C. (2010). *Economic Trends in Enterprise Search Solutions*. Institute for Prospective Technological Studies.

Bernardo, P. (2011). *Padrões de testes automatizados* (Dissertação de Mestrado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, Brasil.

Bisson, T., Patel, Y., & Pasupathy, S. (2012). Designing a fast file system crawler with incremental differencing. In *ACM SIGOPS Operating Systems Review* (3rd ed., Vol. 46, pp. 11–19). New York, USA: ACM.

Bonanomi, R., Silva, W., & Rocha, D. (2012). Efeito da Aplicação do FMEA na Priorização de Riscos de Projetos de Desenvolvimento de Software - Produto. *Estudo & Debate*, *19*(1), 07–23.

Chen, C. L. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*.

Cleverley, P., & Burnett, S. (2015). Retrieving haystacks: a data driven information needs model for faceted search. *Journal of Information Science*, *41*(I), 97–113.

Croft, W. B., Donald Metzler, & Strohman, T. (2015). *Search Engines Information Retrieval in Practice*. Pearson Education, Inc.

Douglas, E. (2014). *JavaScript Furtivo*. Leanpub.

Escada, N. (2012). *csSECURE- Automated Data Discovery and Protection* (Mestrado). Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, Portugal.

Esteso, M. (2015). *Análisis de Arquitecturas de Processado de streaming Big Data* (Dissertação de Mestrado). Universidade politécnica de Madrid - Escola Técnica Superior de Ingenieros de telecomunicación, Madrid.

Ferreira, J. (2013). AGILE approaches in project management.

Findwise. (2014). *Enterprise Search and Findability Report 2014*.

Findwise. (2015). *Enterprise Search and Findability Survey 2015*.

Fortin, M.-F., Côté, J., & Filion, F. (2006). *Fundamentos e Etapas do Processo de Investigação* (Lusodidacta.). Loures.

Gantz, J., & Reinsel, D. (2010). *The Digital Universe Decade – Are You Ready?* IDC.

Garg, N. (2015). *Learning Apache Kafka* (2nd edition.). Packt Publishing.

Germoglio, G. (2010). *Arquitetura de Software*. Creative Commons Attribution 3.0 license.

González, M. (2014). *Extração de Relações Semânticas. Recuirsos, Ferramentas e Estratégias* (PhD Thesis). Departamento de Língua Espanhola, Universidade de Compostela, Santiago de Compostela.

Google. (2006). Enterprise Search Evaluation Guide.

Grant, S., & Schymik, G. (2014). Using Work System Theory to Explain Enterprise Search Dissatisfaction. In *Proceedings of the Information Systems Educators Conference* (Vol. 31). Baltimore, Maryland USA.

Haloi, S. (2015). *Apache Zookeeper Essentials*. Birmingham, UK: Packt Publishing.

Hawking, D. (2004). Chalenges in Enterprise Search. In *Fifteenth Australasian Database Conference (ADC2004)* (Vol. 27). Dunedin, New Zeland.

Hawking, D. (2010). Enterprise Search. In *Modern Information Retrieval* (2nd Ed., pp. 641–684). Pearso Educational.

Hearst, M. A. (1999). User Interfaces and Visualization. In *Modern Information Retrieval*. Essex, England: Addison-Wesley.

Hughes, J., King, V., Rodden, T., & Andersen, H. (1995). *The Role of Ethnography in Interactive Systems Design* (Technical report No. CSEG/8/1995). Lancaster: Computing Deopartment, Lancaster University.

IEEE. (2014). *SWEBOK V3.0 - Guide to the Software Engineering Body of Knowledge*.

IEEE Computer Society. (1998). *IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements Specifications*. New York City:IEEE.

ISO/IEC/IEEE 29119-3. (2013). Software and systems engineering - Software testing - Part 3: Test documentation.

Khabsa, M., Carman, S., Choudhury, S. R., & Giles, C. L. (2012). A Framework for Bridging the Gap Between Open Source Search Tools. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval* (pp. 32–39). Portland, Oregon, USA: Departamento of Computer Science, University of Otago, Dunedin, New Zeland.

Kohn, A., Bry, F., & Manta, A. (2010). Semantic Search on Unstructured Data: Explicit Knowledge through Data Recycling. *International Journal on Semantic Web and Information Systems*, *6*(2).

Koren, J. (2013). *Fulltext Search in the Database and in Texts of Social Networks* (Master Thesis). University of West Bohemia, Pilsen.

Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, *36*(6), 47–56.

Leher, M. (2014). *Building the Business Case for Taxonomy* (White Paper). Denver, Colorado, U.S.A.: Wand, Inc.

Lew, M., Sebe, N., Djeraba, C., & Jain, R. (2006). Content-Based Multimedia Information Retrieval: State of the Art and Challenges. In *ACM Transactions on Multimedia Computing, Communications and Applications* (Vol. 2).

Liu, X., Chen, F., Fang, H., & Wang, M. (2014). Exploiting entity relationship for query expansion in enterprise search. In *Information Retrieval* (3rd ed., Vol. 17, pp. 265–294). Springer.

Liu, X., Fang, H., Chen, F., & Wang, M. (2012). Entity Centric Query Expansion for Enterprise Search. In *CIKM '12 Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1955–1959). Maui, HI, USA: ACM New York, NY, USA.

Li, Y., Liu, Z., & Zhu, H. (2014). Enterprise Search in the Big Data Era: Recent Developments and Open Challenges. In *Proceedings of the VLDB Endowment* (Vol. 7). Hangzhou, China.

Machado, J. (2008). *Recuperação de Informação em Ambientes Semi-Estruturados* (Mestrado). Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal.

Mahanti, A. (2006). Challenges in Enterprise Adoption of Agile Methods – A Survey. *Journal of Computing and Information Technology*, *14*(3), 197–206.

Martins, F. M. (2009). *JAVA6 e Programação Orientada pelos Objetos* (2ª Edição.). Lisboa: FCA - Editora de Informática.

Meier, J. D., Homer, A., Hill, D., Taylor, J., Bansode, P., Wall, L., … Bogawat, A. (2008). *Application Architecture Guide 2.0: patterns & practices* (Microsoft Corporation.).

Middleton, C., & Baeza-Yates, R. (2007). A comparison of open source search engines.

Molková, L. (2011). *Indexing Very Large Text Data* (Master's Thesis). Faculty of Informatics, Masaryk University, Brno, Czech Republic.

Moura, J. (2012). *SESE - Search Quality and Analytics* (Mestrado). Departamento de Engenharia Informática, Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, Portugal.

Nunes, S. (2006). *State of the Art in Web Information Retrieval* (Internal Technical Report). University of Porto.

O'Riordan, A. (2014). Open Search Environments: The Free Alternative to Commercial Search Services. *Information Technology and Libraries*, *33*(2), 45–60.

PengFei (Vincent), L., Thomas, P., & Hawking, D. (2013). Merging algorithms for enterprise search. In *ADCS '13 Proceedings of the 18th Australasian Document Computing Symposium* (pp. 42–49). Queensland University of Technology (QUT) Gardens Point Campus, Brisbane, Queensland, Australia: ACM New York, NY, USA.

Pettersson, F., & Pettersson, N. (2013). *Implementing an enterprise search platform using Lucene.NET* (Master's Thesis). Department of Computer and Information Science, Linköpings universitet, Linköping, Sweden.

Powel, T. (2010). *HTML & CSS: The Complete Reference*. Mc Graw Hill.

Project Management Institute. (2008). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (4th ed.).

Ravikumar, P. (2014). *Enterprise Search Technology Using Solr and Cloud* (Master of Science). Governors State University, University Park, Illinois, USA.

Reinhart, I. (2013). *Enterprise Search Tactics for Expertise Location* (Master of Science). UNIVERSITY OF CALGARY, Calgary, Alberta, Canada.

Russel-Rose, T., Lamantia, J., & Burrell, M. (2011). A Taxonomy of Enterprise Search. Presented at the Enterprise Search Europe, London.

Sanaka, S. (2010). *Faceted Search and Browsing of Indonesian Text Collection Using Shallow Parsing Techniques* (Master of Science). Arizona State University, Tempe, Arizona, USA.

Sanderson, M., & Croft, W. B. (2012). The History of Information Retrieval Research. In *Proceedings of the IEEE* (Vol. 100).

Schwaber, K., & Sutherland, J. (2013). Um guia definitivo para o Scrum: As regras do jogo.

Search Technologies. (2015a). *A Big Data Architecture for Search* (White Paper No. Version 2.0). Search Technologies.

Search Technologies. (2015b). *Enterprise Search Fundamentals* (White Paper No. Version 1.6). Search Technologies.

Semedo, M. (2012). *Ganhos de produtividade e de sucesso de Metodologias Ágeis VS Metodologias em Cascata no desenvolvimento de projectos de software* (Dissertação de Mestrado). Universidade Lusófona de Humanidades e Tecnologias, Lisboa, Portugal.

Sharma, N. (2011). *Enterprise Search Solutions Based on Target Corpus Analysis and External Knowledge Repositories* (M.S. by Research in Computer Science and Engineering). International Institute of Information Technology, Hyderabad - 500 032, INDIA.

Simon, P. (2010). *The Next Wave of Technologies: Opportunities from Chaos*. Hoboken, New Jersey, USA: John Wiley & Sons, Inc.

Softic, S., Rosemberger, M., Zoier, M., Mondelos, K., & Pillinger, E. (2013). A Preliminary Short Survey of State of the Art Enterprise Search Engines for Future Work Place. In *Proceedings of the 9th International Conference on Web Information Systems and Technologies* (pp. 626–629). Aachen. Germany.

Sommerville, I. (2007). *Engenharia de Software* (8ª edição.). São Paulo: Pearson, Addison Wesley.

Stenmark, D., Gårdelöv, F., & Larsson, V. (2015). Why should Organisations Govern Enterprise Search? In *Proceedings of Americas Conference on Information Systems*. Puerto Rico.

Swanson, D. (1961). Information Retrieval: State-of-the-Art. In *Proceedings of the Western Joint Computer Conference* (p. 239). Los Angeles, California.

Taveira, L. (2015). *Monitoramento de Ambientes Computacionais Distribuídos em Tempo Real* (Monografia apresentada como requisito parcial para conclusão do Bacharelado em Engenharia da Computação). Instituto de Ciências Exatas, Departamento de Ciência da Computação, Universidade de Brasília, Brasília.

Tomás, M. (2009). *Métodos ágeis: características, pontos fortes e fracos e possibilidades de aplicação* (IET Working Papers Series No. WPS09/2009). Lisboa: FCT-UNL.

Transparency Market Research. (2014). *Enterprise Search Market (Government and commercial offices, Banking and financial services, Retail, Healthcare and Others) - Global Industry Analysis, Size, Share, Growth, Trends and Forecast 2013 - 2019*.

Turner, V., Gantz, J., Reinsel, D., & Minton, S. (2014). *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things* (White Paper). IDC.

Valentín-Rodríguez, A., Ojeda-Castro, A. M., Alanís-González, M., Márquez-Martínez, E., & Curbelo-Ruiz, A. M. (2015). Critical Factors in the Adoption of Open Source Technologies. *Issues in Information Systems*, *16*(IV), 111–120.

Voyager. (2014). *Measuring the ROI of Enterprise Search*. Voyagersearch.

Wallé, M. (2008). *Basora Enterprise Search A New System for Enterprise Information Retrieval* (MSc. Thesis). Electrical Engineering, Mathematics & Computer Science, Delft University of Technology, Delft, The Netherlands.

White, M. (2012). *Enterprise Search*. O'Reilly Media.

White, M., & Nikolov, S. G. (2013). *Enterprise Search in the European Union: A Techno-economic Analysis* (No. EUR 26000 EN). Luxembourg: European Commission.

Wilhelmsson, F. D., & Eriksson, M. (2013). *Enterprise Search Management Maturity - A model for the assessment of an organization's maturity level within enterprise search management* (Master of Science Thesis). KTH Industrial Engineering and Management, Industrial Management, Stockholm, Sweden.

Woo, J. (2013). Information Retrieval Architecture for Heterogeneous Big Data on Situation Awareness. *International Journal of Advanced Science and Technology*, *59*, 113–122.

Wu, M., Turpin, A., Thom, J., Scholer, F., & Wilkinson, R. (2014). Cost and Benefit Estimation of Experts' Mediation in an Enterprise Search. *Journal of the Association for Information Science and Technology*, pp. 146–163.

ANEXOS

[Esta página foi intencionalmente deixada em branco]

Anexo 1
Lista de fornecedores enterprise search

Company	Country	URL
Active Navigation	UK	http://www.activenav.com
Alcove9	USA	http://www.alcove9.com
Ankiro	Denmark	http://www.ankiro.com
Apache (Lucene/Solr)	Community	http://lucene.apache.org
Attensity	USA	http://www.attensity.com
Attivio	USA	http://www.attivio.com
Autonomy	USA	http://www.autonomy.com
BA-Insight	USA	http://www.bainsight.com
Basis Technology	USA	http://www.basistech.com
Cogito	Italy	http://www.expertsystem.net
Concept Searching	UK	http://www.conceptsearching.com
Constellio	Canada	http://www.constellio.com
Coveo	Canada	http://www.coveo.com
Dieselpoint	USA	http://dieselpoint.com
dtSearch	USA	http://www.dtsearch.com
ElasticSearch	Community	http://www.elasticsearch.org
Elastic Search	Netherlands	http://www.elasticsearch.com
Exalead	France	http://www.exalead.com
Exorbyte	USA	http://www.exorbyte.com
Funnelback	Australia	http://www.funnelback.com
Google	USA	http://www.google.com/services
Inbenta	Spain	http://www.inbenta.com
Intelligenx	USA	http://www.intelligenx.com
Intellisearch	Norway	http://www.intellisearch.com
Intrafind	Germany	http://www.intrafind.de
Lexalytics	USA	http://www.lexalytics.com
LTU	France	http://www.ltutech.com
LucidWorks	USA	http://www.lucidworks.com
Mark Logic	USA	http://www.marklogic.com
MaxxCAT	USA	http://www.maxxcat.com
Mindbreeze	Austria	http://www.mindbreeze.com/en
Ontolica	Denmark	http://www.surfray.com
OpenSearchServer	France	http://www.open-search-server.com
Perfect Search	USA	http://www.perceptivesoftware.com
Perceptive Software	USA	http://www.perfectsearchcorp.com
Polyspot	France	http://www.polyspot.com
Q-Sensei	USA	http://www.qsensei.com
Ravn	UK	http://www.ravn.co.uk
Recommind	USA	http://www.recommind.com
SchemaLogic	USA	http://www.schemalogic.com
SearchBlox	USA	http://www.searchblox.com
SearchDaimon	Sweden	http://www.searchdaimon.com
Simplexo	UK	http://www.simplexo.com
Sinequa	France	http://www.sinequa.com
SLI Systems	New Zealand	http://www.sli-systems.com
Smart Logic	UK	http://www.smartlogic.com
Sphinx	Community	http://sphinxsearch.com
Synaptica	USA	http://www.synaptica.com
Temis	France	http://temis.com
Teragram	USA	http://www.teragram.com/oem
TeraText	USA	http://www.teratext.com
Terrier	UK	http://terrier.org
Thetus	USA	http://thetus.com
Thunderstone	USA	http://www.thunderstone.com
Vivisimo	USA	http://www.vivisimo.com
Wand	USA	http://wandinc.com
Xapian	Commuity	http://xapian.org
X1 Technologies	USA	http://www.x1.com
ZyLab	USA	http://zylab.com

[Esta página foi intencionalmente deixada em branco]

Anexo 2
Modelo de questionário

Pesquisa Empresarial

1. **Qual a sua função na Critical Software?**
   Marcar apenas uma oval.

   - ◯ Business Development - Business Development
   - ◯ Business Development - Engineering Pre Sales
   - ◯ Business Development - Marketing
   - ◯ Delivery - Engineering
   - ◯ Delivery - Project Management
   - ◯ Delivery - Delivery Management
   - ◯ Support Operations - Quality
   - ◯ Support Operations - Finance
   - ◯ Support Operations - Innovation & Knowledge
   - ◯ Support Operations - Information Technologies
   - ◯ Support Operations - Human Resources
   - ◯ Support Operations - Operations

2. **Qual a sua categoria na Critical Software?**
   Marcar apenas uma oval.

   - ◯ C1
   - ◯ C2
   - ◯ C3
   - ◯ C4
   - ◯ C5

3. **Qual ou quais a/as língua que utiliza com mais frequência na pesquisa de informação empresarial'**
   Marcar tudo o que for aplicável.

   - ☐ Português
   - ☐ Inglês
   - ☐ Francês
   - ☐ Espanhol
   - ☐ Outra: .....

4. **Qual ou quais os elementos que, em sua opinião, são mais relevantes na pesquisa de informação empresarial?**
Marcar apenas uma oval.

- ◯ tipo de documento (memorando, documento de qualidade, acta de reunião, relatório financeiro, ...)
- ◯ tamanho do documento
- ◯ documentos revistos por outros
- ◯ data
- ◯ título
- ◯ sumário
- ◯ linguagem
- ◯ Outra: .....

5. **Qual o tipo de filtro que considera mais relevante na pesquisa de informação empresarial?**
Marcar apenas uma oval.

- ◯ tipo de documento (memorando, documento de qualidade, acta de reunião, relatório financeiro, ...)
- ◯ keyword
- ◯ data
- ◯ linguagem
- ◯ documentos revistos por outras pessoas
- ◯ Outra: .....

6. **Qual ou quais dos seguintes critérios, de relevância dos resultados obtidos, em sua opinião, é uma mais valia?**
Marcar apenas uma oval.

- ◯ completude (o documento contem todos os termos de pesquisa ou apenas alguns?)
- ◯ proximidade (os documentos com as palavras pesquisadas com menos afastamento são considerados mais relevantes)
- ◯ densidade (percentagem de vezes em que aparecem as palavras pesquisadas no documento)
- ◯ metadata (termos de pesquisa no título, abstract, ...)

7. **Que tipo de informação usa com mais frequência como critério de pesquisa?**
Marcar apenas uma oval.

- ◯ linguagem natural (pesquisa por frase)
- ◯ palavras-chave simples (Pesquisa por palavra única)
- ◯ palavras-chave composta (Várias palavras sem operadores)
- ◯ operadores booleanos (NOT, AND, OR, ...)
- ◯ Outra: .....

8. **Qual o tipo de documentos que pesquisa com mais frequência?**
Marcar apenas uma oval.

- ◯ pdf
- ◯ doc(x)
- ◯ ppt(x)
- ◯ xls(x)
- ◯ plain
- ◯ Outra: .....

9. **Em sua opinião o apoio de um mecanismo que sugira termos de pesquisa em função dos caracteres introduzidos (sugestão automática) é importante?**
Marcar apenas uma oval.

	1	2	3	4	5	
Pouco importante	◯	◯	◯	◯	◯	Muito importante

10. **Em sua opinião, no âmbito da pesquisa de informação empresarial, a antiguidade dos documentos é relevante?**
Marcar apenas uma oval.

	1	2	3	4	5	
Pouco Relevante	◯	◯	◯	◯	◯	Muito Relevante

11. **Se tiver alguma sugestão que possa ser útil para os requisitos de um sistema de pesquisa de informação empresarial, agradeço que a transmita no espaço abaixo.**

.....

.....

.....

.....

.....

Com tecnologia
Google Forms