



Relatório Final
Mestrado em Engenharia Informática

Wizdee Discovery Colaboração e Personalização

Tiago José Pereira dos Santos Mateus

Orientadores
Doutor Bruno Antunes (Wizdee)
Doutor Paulo de Carvalho (DEI)

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

2015

Relatório Final
Mestrado em Engenharia Informática

Wizdee Discovery

Colaboração e Personalização

Tiago José Pereira dos Santos Mateus

Orientadores

Doutor Bruno Antunes (Wizdee)

Doutor Paulo de Carvalho (DEI)

Jurados

Doutor Marco Vieira

Doutor Paulo de Carvalho

Doutor Rui Pedro Paiva

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

2015

Resumo

As abordagens de *Business Intelligence* (BI) são caracterizadas pelo envolvimento e desenvolvimento de sistemas de armazenamento de dados complexos (*Data Warehouses*) cujo custo inicial e de manutenção são elevados, necessitando de pessoas técnicas para a sua criação e manutenção. De forma a contrariar estas necessidades, surge o *Wizdee Discovery*, produto desenvolvido pela Wizdee, o qual torna mais fácil a exploração dos dados em sistemas de BI, através da sua caixa de pesquisa. Nesta caixa, os utilizadores do sistema, através do uso de linguagem natural, conseguem pesquisar e explorar os seus dados.

Nesta tese, são apresentados duas funcionalidades que trazem valor acrescido ao produto: colaboração e personalização. A primeira prende-se com a possibilidade de os utilizadores poderem colaborar entre si na análise dos seus dados e poderem partilhar essa informação, tanto a nível interno do sistema como a nível externo, sendo o caso da partilha para redes sociais ou ferramentas de produtividade, como a *Dropbox*. Por outro lado, a personalização do *Wizdee Discovery* pretende que os resultados sejam adaptados a cada utilizador segundo as interações que este tem com a plataforma. Nos sistemas de linguagem natural, um dos maiores problemas é a ambiguidade associada. Desta forma, com a introdução da personalização é esperado que exista uma redução deste problema. Para isto, é necessário a introdução de contexto, o qual permite verificar quais os dados mais importantes para o utilizador no momento da pesquisa dos seus dados.

As funcionalidades colaborativas implementadas são usadas por alguns competidores da Wizdee, sendo a partilha para ferramentas de produtividade o elemento diferenciador. Noutra vertente, as funcionalidades de personalização quando aplicadas a motores de pesquisa são exploradas por diversos investigadores, existindo as mais variadas abordagens. Este trabalho segue uma abordagem baseada no contexto, através do qual é possível ir moldando a importância que os dados têm para o utilizador em determinado momento. Este contexto tem como base um conjunto de pesos associados às estruturas dos dados, como tabelas e colunas, e são estes que vão sendo alterados segundo a interação do utilizador com o sistema. Para isto ser possível, são também usadas abordagens baseadas em algoritmos genéticos, os quais são usados para melhorar o processo de avaliação de cada interpretação do sistema às pesquisas introduzidas pelo utilizador.

Parte das funcionalidades implementadas foram testadas pela equipa de testes da empresa e a outra parte foi testada pelo estagiário. Os resultados obtidos permitiram verificar que a abordagem seguida tem um impacto positivo na colaboração entre utilizadores e na resposta do sistema às pesquisas introduzidas por utilizadores com diferentes contextos. Além disso, os resultados mostram que os utilizadores consideram útil a introdução do contexto de forma a melhorar a obtenção dos resultados esperados.

Palavras-Chave: Colaboração, Partilha, Personalização, Contexto, Algoritmos Genéticos

Abstract

Business Intelligence's typical approach is based on the development of complex data information systems (Data Warehouses). Normally, it requires technical people to create and manage the infrastructure and has high initial and maintenance costs. In an attempt to overcome this, Wizdee launched Wizdee Discovery, a product which aims to facilitate data exploitation in Business Intelligence (BI) systems, by letting users explore data through a search box.

With this tool users are able to search and exploit their data by using natural language. In this thesis, two product functionalities are highlighted, due to the added value created to the company's product: collaboration and personalization. The first one relates to the possibility for users to be able to collaborate between them regarding their data analysis and share that information, both internally and externally, as is the case of social networks or productivity tools, like Dropbox.

On the other hand, Wizdee Discovery personalization aims to adapt the results to each user according to his interaction with the platform. Concerning natural language systems, the related ambiguity is one of the major problems. With the introduction of the personalization feature, we expect to achieve an improvement in this matter. In order for this to happen, it is necessary to introduce context, which allows to identify which data is more relevant to the user.

The implemented collaborative functionalities are also used by some of the Wizdee's competitors, being the external share to productivity tools one of the company's main points of difference. In another level, the personalization functionalities applied to search engines have already been exploited by many researchers, leading to several perspectives.

This paper follows an approach based on context, through which it is possible to shape the importance that the data has for the user. This context is based on a set of weights associated to the structure of data, like tables and columns. These weights will change according to the interaction of the user with the system. In order for this to be possible, approaches based on genetic algorithms are also used. The goal is to improve the evaluation process of the interpretation that the system makes for each query.

Part of the the implemented functionalities were tested by the company's quality team and the other part was tested by the intern. The results allowed to verify that both approaches had a positive impact in the collaboration (between users) and in the personalization (system response to the queries introduced by users with different contexts). Moreover, the survey results show that the users consider that the introduction of context improves the achievement of the expected results.

Keywords: Collaboration, Share, Personalization, Context, Genetic Algorithms

Agradecimentos

Esta tese representa o alcançar de mais uma meta da minha vida, o qual não seria possível sem a intervenção de algumas pessoas. Desta forma, resta-me agradecer àquelas que me ajudaram ao longo deste percurso.

Em primeiro lugar queria deixar um agradecimento especial aos meus orientadores, Doutor Bruno Antunes e Doutor Paulo de Carvalho, os quais se revelaram incansáveis no acompanhamento do meu trabalho e estiveram sempre prontos a dar o seu contributo. Ainda na qualidade de orientador, mas não oficial, queria agradecer ao Doutor Paulo Gomes. Numa primeira instância por me ter dado a oportunidade de trabalhar na Wizdee e numa segunda instância pelo constante apoio na orientação do meu estágio.

Já agradei a oportunidade de poder trabalhar numa empresa como a Wizdee. Ainda não agradei foi à equipa que encontrei. Uma equipa que desde cedo se mostrou disponível para me ajudar e me fez crescer tanto a nível pessoal como profissional. Hoje, considero esta equipa como uma família e também a Wizdee como uma segunda casa. Um obrigado enorme a esta equipa que é pequena em número mas enorme no seu valor.

Tenho que também agradecer a todos os meus amigos, aqueles que me moldaram e tornaram a pessoa que sou hoje. Todas as nossas desavenças e alegrias, todos os nossos choros e risos, tudo valeu a pena. Cada um contribuiu à sua maneira, uns mais outros menos, mas o valor do seu contributo total é inestimável.

Por último e não menos importante, antes pelo contrário, queria agradecer aos meus pais. A formação que me deram, todas as regalias que me permitiram, todos os esforços que fizeram por mim para nunca me faltar nada e também por todo o amor transmitido. Sinceramente, espero que este meu percurso vos tenha deixado orgulhosos e represente uma pequena retribuição por tudo o que fizeram e continuam a fazer por mim.

Um enorme obrigado a todos vós.

Índice

Capítulo 1: Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	1
1.3 Abordagens	2
1.4 Contribuições	3
1.5 Estrutura do Documento	4
Capítulo 2: Conceitos Base e Trabalhos Relacionados	5
2.1 Colaboração e Partilha	5
2.1.1 Abordagens	6
2.1.2 Competidores	7
2.2 Personalização	12
2.2.1 Abordagens	12
2.2.2 Trabalhos Relacionados	25
Capítulo 3: Abordagem	29
3.1 Enquadramento	29
3.2 Objetivos	31
3.2.1 Colaboração	31
3.2.2 Ordenação de Cenários	32
3.2.3 Personalização	33
3.3 Análise de Requisitos e Restrições	33
3.3.1 Requisitos Funcionais	33
3.3.2 Requisitos de Desempenho	37
3.3.3 Requisitos de Qualidade	37
3.3.4 Requisitos Linguísticos	37
3.3.5 Restrições Tecnológicas	38
3.4 Arquitetura	38
3.4.1 Camada de Dados	39
3.4.2 Camada de Negócio	39
3.4.3 Camada de Apresentação	40
3.5 Componentes	40
3.5.1 Componentes Alterados	40
3.5.2 Componentes Criados	41
3.5.3 Componentes Usados	41
3.6 Análise de Riscos	42
Capítulo 4: Implementação	45
4.1 Colaboração	45
4.1.1 Partilha Interna	45
4.1.2 Partilha Externa	47

4.1.3	Suporte para Comentários	50
4.1.4	Suporte para Notificações	51
4.2	Ordenação de Cenários	52
4.2.1	Adaptação do Algoritmo Genético	52
4.2.2	Otimização do Algoritmo Genético	53
4.2.3	Aplicação	54
4.3	Personalização	55
4.3.1	Contexto de Longo Prazo	55
4.3.2	Contexto de Curto Prazo	60
4.3.3	Aplicação	60
Capítulo 5: Testes e Validação	63
5.1	Colaboração e Partilha	63
5.1.1	Testes Unitários	63
5.1.2	Testes de Aceitação	63
5.2	Ordenação de Cenários	64
5.2.1	Testes Automáticos	64
5.2.2	Análise de Resultados	65
5.3	Personalização	65
5.3.1	Testes Quantitativos	65
5.3.2	Testes Qualitativos	70
5.3.3	Análise de Resultados	71
Capítulo 6: Metodologia e Planeamento	73
6.1	Metodologia	73
6.2	Planeamento	74
6.2.1	Primeiro Semestre	74
6.2.2	Segundo Semestre	75
Capítulo 7: Conclusões	79
7.1	Objetivos	79
7.2	Abordagem	79
7.3	Resultados	80
7.4	Contribuições	81
7.5	Trabalho Futuro	81
Apêndice A: Referência do API de Colaboração	83
Apêndice B: Testes Unitários de Colaboração e Partilha	93
Apêndice C: <i>Features</i> do Algoritmo Genético	99
Bibliografia	101

Lista de Figuras

2.1	Geração de <i>links</i> por parte da <i>InVision App</i>	6
2.2	Ilustração da rede neuronal mais simples.	17
2.3	Processo evolutivo usado nos algoritmos genéticos (Cunha et al., 2012).	19
2.4	Seleção por roleta.	20
2.5	Cruzamento por um ponto.	21
2.6	Personalização de resultados a partir dos favoritos (rae Kim and Chan, 2005).	26
2.7	Personalização de resultados a partir dos módulos de perfil de utilizador e do seu padrão de navegação (Liu et al., 2010).	27
3.1	Organização dos livros.	29
3.2	Barra de pesquisa com <i>autocomplete</i>	30
3.3	Visão geral dos objetivos da colaboração.	32
3.4	Visão geral dos objetivos da personalização.	34
3.5	Arquitetura do sistema.	38
3.6	Matriz de exposição de riscos.	42
4.1	Diagrama da base de dados.	46
4.2	Partilha de <i>widgets</i> entre grupos.	46
4.3	Partilha de <i>widgets</i> entre utilizadores.	47
4.4	Partilha de <i>widget</i> por email.	48
4.5	Diagrama de autenticação <i>OAuth</i>	49
4.6	Organização de tarefas no Trello.	50
4.7	Notificações do utilizador.	51
4.8	Cruzamento por um ponto.	52
4.9	Processo para a implementação das <i>features</i>	54
5.1	Testes automáticos com todas as 500 pesquisas.	64
5.2	Testes quantitativos da ordenação de cenários por contexto simulado.	66
5.3	Testes quantitativos do <i>autocomplete</i> por contexto simulado.	67
5.4	Comparação dos tempos de geração de cenário entre a abordagem antiga e a abordagem nova.	67
5.5	Testes quantitativos da ordenação de cenários pelo exercício com utilizadores.	68
5.6	Testes quantitativos do <i>autocomplete</i> pelo exercício com utilizadores.	69
6.1	Diagrama de Gantt respeitante ao primeiro semestre.	77
6.2	Diagrama de Gantt respeitante ao segundo semestre.	78

Lista de Tabelas

2.1	Ferramentas de BI e respectivas funcionalidades.	11
3.1	Requisitos Funcionais para Colaboração.	34
3.2	Requisitos Funcionais da Partilha Externa.	35
3.3	Requisitos Funcionais para Personalização.	36
3.4	Requisitos Linguísticos.	38
3.5	Restrições Tecnológicas.	38
3.6	Lista de Riscos e respetivo Grau de Risco.	42
4.1	Fontes de informação consideradas no contexto de longo prazo.	55
4.2	Mapeamento temporário das diferenças.	57
4.3	Mapeamento temporário depois da primeira comparação.	58
4.4	Mapeamento temporário depois da segunda comparação.	58
4.5	Mapeamento temporário depois da terceira comparação.	58
4.6	Mapeamento temporário depois da quarta comparação.	59
4.7	Valores finais da variação de contexto e respetiva normalização.	59
4.8	Novos pesos para cada estrutura.	59
4.9	Eventos do contexto de curto prazo.	60
5.1	Estatísticas associadas aos testes com contexto simulado.	68
5.2	Estatísticas associadas aos testes realizados nos exercícios com utilizadores.	69
A.1	Referência do API de Colaboração.	84

Acrónimos

AG *Algoritmo Genético*

API *Application Programming Interface*

BI *Business Intelligence*

HTML *HyperText Markup Language*

MD5 *Message Digest 5*

ML *Machine Learning*

PDF *Portable Document Format*

PNG *Portable Network Graphics*

SQL *Structured Query Language*

SVG *Scalable Vector Graphics*

SVM *Support Vector Machines*

UX *User Experience*

Capítulo 1

Introdução

O presente documento apresenta o trabalho realizado por Tiago Mateus, durante o ano académico 2014/2015, no estágio do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologias da Universidade de Coimbra. O estágio decorreu na *Wizdee*¹, uma empresa criada em 2009 como um *spin-off* do grupo *Cognitive and Media Systems*² do centro de investigação CISUC³, especializada na área de web semântica, processamento de linguagem natural e *Business Intelligence* (BI), onde o estagiário se encontra integrado na equipa de desenvolvimento. O estágio foi orientado pelo Doutor Bruno Antunes na *Wizdee* e pelo Professor Paulo de Carvalho por parte do Departamento de Engenharia Informática.

1.1 Enquadramento

A abordagem típica de BI (Kimball and Ross, 2013) envolve o desenvolvimento de sistemas de armazenamento de dados complexos (*Data Warehouses*) cujo custo inicial e de manutenção são elevados, necessitando de pessoas técnicas para a sua criação e manutenção. O *Wizdee Discovery*, produto desenvolvido pela *Wizdee*, surge como forma de tornar mais fácil a criação e exploração de dados em sistemas de BI. O conceito principal do *Wizdee Discovery* é a pesquisa como forma principal de interação com o utilizador, sendo através da caixa de pesquisa que é realizada a pesquisa e exploração dos dados recorrendo a linguagem natural, ou seja, linguagem humana. Através do uso de tecnologias da web semântica (Antoniou and Harmelen, 2008) e processamento de linguagem natural, o *Wizdee Discovery* gera uma resposta, a qual é uma interpretação da pesquisa que o utilizador introduziu. Desta maneira, qualquer tipo de utilizador pode ver as suas questões de negócio facilmente respondidas e gerar relatórios com as análises pretendidas, não estando dependente de uma equipa técnica, como anteriormente foi referido.

1.2 Objetivos

O presente estágio incidiu em dois objetivos distintos: desenvolvimento de um conjunto de funcionalidades de colaboração e partilha entre utilizadores do *Wizdee Discovery*; e desenvolvimento de funcionalidades de personalização do *Wizdee Discovery*.

Colaboração e Partilha. Um dos aspetos importantes nas soluções de BI é a possibilidade de os utilizadores conseguirem partilhar informação e colaborar. É importante

¹ *Wizdee* - <http://wizdee.com>

² *Cognitive and Media Systems* - <http://www.uc.pt/en/fctuc/ID/Organization/CMS/>

³ CISUC - <http://www.uc.pt/en/fctuc/ID/cisuc>

que a informação que conseguem extrair dos dados seja facilmente partilhada e discutida com outros utilizadores. Neste sentido, um dos objetivos do estágio é a implementação de diversas funcionalidades de colaboração capazes de ajudar os utilizadores na construção e partilha dos seus relatórios. Esta parte apresenta claramente uma vertente mais forte de engenharia. Desta maneira, as funcionalidades implementadas permitirão a partilha de relatórios por *link*, email, redes sociais e ferramentas de produtividade⁴. Ainda neste âmbito, o suporte para comentários e notificações também será introduzido no sistema, o qual permite a adição de comentários nos relatórios e a notificação ao utilizador que algo foi partilhado com ele.

Personalização. Outro dos objetivos principais do estágio é a capacidade de personalização do sistema, estando este objetivo mais virado para uma vertente de investigação. Cada utilizador tem um contexto próprio e necessidades específicas, as quais devem ser tidas em conta para o sistema de pesquisa ser mais eficaz. O objetivo é o *Wizdee Discovery* ser capaz de ter um grau de personalização, adaptando-se ao utilizador. Para isso, o sistema tem que estar dotado de algoritmos capazes de modelizar o contexto do utilizador, o qual deve ser modelizado de acordo com as interações que realiza com a plataforma. Este contexto de utilizador irá ser usado para facilitar o acesso à informação relevante, adaptando as respostas geradas pelo *Wizdee Discovery* ao utilizador.

Um exemplo de um melhoramento em que a personalização vai dar o seu contributo é na desambiguação de uma pesquisa. Com o contexto do utilizador traçado e existindo ambiguidade numa pesquisa, a escolha da solução gerada irá ser a mais adequada àquele utilizador específico. Por exemplo, existindo um atributo denominado de "Cidade" em duas tabelas diferentes "Utilizador" e "Contacto", se o utilizador toma sempre a ação de ver este atributo na tabela "Contacto", o sistema deve ser capaz de construir um contexto de utilizador, de modo a que as próximas respostas a uma pesquisa com "Cidade" se refiram a "Cidade" da tabela "Contacto".

1.3 Abordagens

As abordagens propostas para o cumprimento dos objetivos propostos encontram-se nos parágrafos seguintes.

Colaboração e Partilha. Pertencendo a uma vertente mais de engenharia, a abordagem seguida passa pela gestão da base de dados do *Wizdee Discovery* e pela alteração dos respetivos comandos, de maneira a que os utilizadores tenham acesso às partilhas que com eles foram realizadas e também a comentários e notificações. Na partilha externa dá-se a preferência ao uso de bibliotecas que já lidam com o processo de autenticação e partilha para essas ferramentas. Caso não exista uma biblioteca capaz de realizar essas operações para uma determinada ferramenta, então os métodos necessários são implementados para fazer uso da API disponibilizada por cada uma das ferramentas.

Personalização. Na análise de trabalhos relacionados foram encontradas as mais diversas formas de personalização, as quais foram propostas por diversos investigadores. No entanto, e tendo a empresa uma abordagem anterior que fazia uso de algoritmos genéticos, a escolha recai sobre estes. Através da otimização deste tipo de algoritmos é melhorada a interpretação que o sistema realiza às pesquisas introduzidas pelo utilizador. Posterior-

⁴Ferramentas de Produtividade - Ferramentas que permitem ao utilizador organizar o seu dia a dia. Estas possibilitam a criação de notas, a marcação de eventos, entre outras. Como exemplo surge a ferramenta da *Dropbox* (<http://www.dropbox.com>)

mente, e seguindo outra abordagem, é introduzido o contexto. Para isto, cada estrutura, como tabelas e colunas, é associada a cada utilizador. A relação é feita através da associação de pesos a essas estruturas, sendo que o peso é tanto maior quanto maior for a importância para o utilizador. Estes pesos são aqueles que vão sendo moldados através da interação do utilizador com a plataforma.

Estas abordagens são implementadas segundo uma metodologia de desenvolvimento denominada de *Scrum* (Rising and Janoff, 2000), a qual é descrita com maior detalhe no capítulo 6.

1.4 Contribuições

As contribuições principais neste estágio prendem-se com a introdução de funcionalidades que permitem a colaboração e partilha de relatórios e, também, com a personalização do sistema. Estas contribuições são descritas nos seguintes pontos:

- **Funcionalidades.** A nível de funcionalidades foram implementadas as descritas nos parágrafos seguintes:
 - **Partilha.** Os utilizadores podem partilhar entre si os dados que quiserem. Podem ainda partilhar esses mesmos dados para o exterior, como por email, *link*, redes sociais ou ferramentas de produtividade. Através destas últimas é possível que o utilizador ganhe tempo, não tendo que andar a trocar de aplicação sempre que deseja adicionar uma nota (*Evernote*⁵) ou marcar um evento no seu calendário (*Google Calendar*⁶).
 - **Colaboração.** Os utilizadores são capazes de colaborar entre si, podendo trabalhar sobre os mesmos dados, conseguindo discutir sobre estes através da realização de comentários sobre esses dados. Para o utilizador saber que algo foi partilhado com ele pode também fazer uso do sistema de notificações.
 - **Melhoria na Avaliação das Interpretações do Sistema.** O sistema encontra-se dotado com uma avaliação mais complexa das interpretações do sistema às pesquisas que o utilizador introduz, levando a uma avaliação mais detalhada e fiável.
 - **Personalização.** Graças ao aumento da complexidade na avaliação de cada pesquisa, é possível a introdução de contexto do utilizador. Assim, o sistema consegue através das interações do utilizador com a plataforma analisar quais os dados mais importantes para o utilizador e assim resolver muitos problemas de ambiguidade, conseguindo que as sugestões de pesquisa e as interpretações do sistema sejam adaptadas a cada utilizador.
- **Modelo Computacional de Personalização.** Modelo computacional novo para a personalização de sistemas de *BI*, onde as suas pesquisas são baseadas em estruturas, como tabelas e colunas.
- **Experimentação sobre Personalização.** Realização de várias experimentações na área da personalização quando aplicada em sistemas como o *Wizdee Discovery*.
- **Relatório de Estágio.** O presente documento.

⁵*Evernote* - <https://www.evernote.com>

⁶*Google Calendar* - <https://www.google.com/calendar>

1.5 Estrutura do Documento

O presente documento possui a seguinte estrutura:

- Conceitos Base e Trabalhos Relacionados (2) - Este capítulo pode ser dividido em dois grande tópicos:
 - Colaboração (2.1) - Descreve os conceitos base bem como faz uma análise dos competidores, apresentando algumas das abordagens que são usadas.
 - Personalização (2.2) - Apresenta os principais conceitos de personalização e algumas abordagens que são seguidas. Por último, é apresentado um conjunto de trabalhos relacionados.
- Abordagem (3) - Começa com um enquadramento do produto, descrevendo algumas das suas funcionalidades. De seguida são apresentados os objetivos do estágio bem como uma análise de requisitos. Posteriormente, é apresentada a arquitetura do sistema e são apresentados os componentes relevantes para o desenvolvimento do trabalho. Por último, é feita uma análise dos riscos.
- Implementação (4) - Descreve as implementações realizadas no âmbito da colaboração e personalização.
- Testes e Validação (5) - Descreve os processos de teste usados para a validação das funcionalidades implementadas.
- Metodologia e Planeamento (6) - É apresentada a metodologia de desenvolvimento seguida pela empresa bem como o planeamento que foi seguido.
- Conclusões (7) - Por último, o documento é terminado com um resumo e apresentação das conclusões.

Capítulo 2

Conceitos Base e Trabalhos Relacionados

Este capítulo descreve os conceitos que estão na base deste trabalho, faz uma análise dos competidores e aborda também diversos trabalhos relacionados. Como já foi referido, o estágio pode ser categorizado em duas vertentes, uma com maior componente de engenharia e a outra de investigação. Na primeira, a informação recolhida está mais relacionada com as técnicas usadas pelos competidores, não existindo uma parte respeitante aos trabalhos relacionados. Por outro lado, a segunda parte carece de informação disponível sobre os processos e algoritmos usados pelos competidores, sendo por isso focados os trabalhos relacionados.

A estrutura deste capítulo é iniciada com a secção de colaboração e partilha, na qual é dada uma visão geral do seu conceito, bem como as suas vantagens e aplicações. Ainda na mesma, é feita uma análise dos competidores, referindo algumas das abordagens usadas por estes. De seguida, vão ser apresentados o conceito de personalização, as suas vantagens e os tipos existentes. As abordagens seguidas serão também descritas, seguindo-se uma análise de diversos trabalhos relacionados.

2.1 Colaboração e Partilha

Uma organização é composta por diversos departamentos que, juntos, tentam atingir um objetivo final comum, apesar das suas funções serem bastante diferentes. É neste espírito de cooperação que entra o conceito de colaboração. Esta pode ser definida como "*agir com outrem para a obtenção de determinado resultado*"¹. Devido a esta necessidade de cooperação e partilha, diversas ferramentas de BI implementam, nos seus sistemas, soluções que permitem a colaboração entre os diferentes utilizadores.

Com este tipo de funcionalidade, a interação entre equipas é facilitada, permitindo uma análise de dados eficiente e a troca de informação e ideias. As aplicações dos sistemas colaborativos são várias: o envio de um relatório para um superior, a análise em conjunto de um determinado gráfico ou a tomada de uma decisão com base no tipo de dados que está a ser analisado. Tudo isto permite um aproximar de todos os membros de uma organização.

De seguida, irão ser referidas algumas das soluções usadas por alguns dos competidores e, posteriormente, é realizada uma correspondência entre alguns destes e a respetiva abordagem seguida.

¹Dicionário da Priberam - www.priberam.pt

2.1.1 Abordagens

Segue-se uma lista das abordagens de colaboração e partilha usadas por outras ferramentas de BI. Não pretendendo ser uma lista exaustiva, mas sim uma referência às principais técnicas usadas.

Geração de *Links*

Existem várias ferramentas que permitem a geração de um *link*, o qual pode ser posteriormente enviado por email ou através de outro canal de comunicação. Este pode ainda ter associado a si uma propriedade que refere o seu tempo de validade, findo o qual o *link* não estará mais disponível para ser consultado.

Este é um dos tipos de partilha usado pela *Dropbox*, em que o utilizador ao pretender partilhar determinada pasta ou ficheiro, clica na opção partilha e, de seguida, é gerado um *link*. Com este *link*, e usando o meio de comunicação da sua preferência, o utilizador pode enviar para o destinatário pretendido e assim permitir o acesso aos dados partilhados.

Uma das maiores preocupações neste tipo de partilha é a falta de segurança, pois quem tiver acesso ao *link* poderá consultar a informação associada. E em ferramentas de BI, este problema é algo que deve ser tido em conta, pois muita da informação deve permanecer de acesso restrito.

Algumas ferramentas permitem ainda adicionar comentários e anotações aos dados que foram partilhados e assim tornar a comunicação entre os utilizadores mais fácil.

Um exemplo desta abordagem pode ser visualizada na figura 2.1.



Figura 2.1: Geração de *links* por parte da *InVision App*.

Geração de HTML

Nesta abordagem, o processo é semelhante ao usado na geração do *link*, mas neste caso é possível a geração de código HTML. Este código permite gerar uma representação visual dos dados, sendo principalmente usado na partilha de gráficos. Posteriormente, o destinatário poderá embeber o HTML na sua página web ou noutra suporte que o permita.

Partilha para Redes Sociais

Neste género de colaboração usam-se as redes sociais como um meio de comunicação. Cada vez mais são as ferramentas que permitem a partilha direta dos seus dados para as redes sociais. Muitas são as páginas web que têm esta funcionalidade. As próprias redes sociais disponibilizam-na de uma maneira facilitada, através dos seus serviços de API.

Partilha para Ferramentas de Produtividade

Existem diversas ferramentas que são orientadas para a produtividade, eficácia e organização do dia a dia. Seja para realizar anotações, marcar eventos ou adicionar tarefas, este

tipo de ferramentas permite a realização destas ações e muitas outras. Como exemplo, temos o *Evernote*², a *Dropbox*, o *Trello*³, entre muitas outras.

Na área de BI, este tipo de ações podem surgir no seguimento da análise de determinados dados, pelo que o utilizador pode ter a necessidade de tirar apontamentos ou marcar uma reunião para debater o estado dos seus negócios. Com este objetivo, a ferramenta de BI oferece ao utilizador a possibilidade de ligar a sua ferramenta de produtividade ao sistema. Desta maneira, o utilizador pode realizar a ação que pretende sem que seja obrigado a sair da ferramenta de BI.

Partilha Interna

Na partilha interna não é utilizada qualquer aplicação externa à ferramenta. Esta abordagem permite que sejam partilhados dados entre os utilizadores do sistema, tendo como único requisito a obrigatoriedade de todos os utilizadores envolvidos estarem registados no sistema. Neste tipo de partilha, algumas ferramentas fornecem ainda a possibilidade de adicionar comentários e anotações sobre os dados que foram partilhados.

Este tipo de partilha é o que pode ser visualizado tanto na *Dropbox* como no *GoogleDocs*⁴, possibilitando a partilha de pastas e/ou ficheiros através da simples introdução do destinatário pretendido. Esta partilha pode ser de diferentes tipos, podendo dar aos outros utilizadores diferentes tipos de permissões, incluindo escrita e/ou leitura.

Conversação ("Chat")

A ferramenta permite a discussão em tempo real sobre determinados objetos. Para isso, surge uma janela em estilo de *chat* que permite a conversação e o anexo de diversos ficheiros. Mais uma vez, este é um dos serviços que o *GoogleDocs* possibilita. Ao ser editado um documento, e estando este partilhado com outro utilizador, uma janela é exibida, onde os dois podem discutir a edição do documento.

Segurança na Partilha

Sendo os sistemas de BI utilizados essencialmente em ambiente empresarial, existem regras que restringem o acesso à informação a determinadas pessoas ou grupos de pessoas. Para isso, são implementadas medidas de segurança que apesar da partilha dos dados com determinado utilizador, garantem que este apenas veja os dados a que tem acesso.

2.1.2 Competidores

Esta secção foca-se nos competidores ligados à área de BI, apresentando as funcionalidades de partilha que estes sistemas possuem. É iniciada com uma breve descrição das ferramentas analisadas, fazendo, posteriormente, uma análise comparativa onde é feita uma correspondência entre as ferramentas e as técnicas utilizadas.

Ferramentas de BI

As ferramentas analisadas são aquelas que mais informação disponibilizam publicamente. A listagem seguinte pretende dar uma breve descrição de cada uma, mencionando as técnicas utilizadas.

² *Evernote* - <http://www.evernote.com>

³ *Trello* - <http://www.trello.com>

⁴ *GoogleDocs* - <https://docs.google.com>

- **Panorama** - O seu *software* - *Panorama Necto* - permite realizar análises personalizadas, intuitivas e interativas, as quais são visualizadas em formato de gráfico.

Esta permite a colaboração entre os utilizadores, utilizando técnicas de geração de *link*, partilha interna e conversação. Na partilha interna, os utilizadores conseguem adicionar anotações, onde conseguem realçar os dados a serem focados.

- **ArcPlan** - *Software* de BI que permite conectar diferentes fontes de dados e fornece ao utilizador a possibilidade de analisar os seus dados, através da visualização gráfica.

Em termos colaborativos, esta ferramenta possibilita a partilha interna, onde os utilizadores conseguem adicionar comentários e pontuações sobre os dados partilhados. É também permitido a adição de anotações e a geração de um *link*.

- **Tableau** - O *Tableau Software* possibilita a conexão com os dados do utilizador bem como a sua respetiva visualização.

Como ferramenta colaborativa permite a geração de um *link*, bem como de HTML e também a partilha para as redes sociais - Facebook e Twitter. Neste último tipo de partilha, a adição de anotações e filtros é também possível.

- **BiTeamWork** - Produto que oferece uma rápida integração em serviços de BI, o qual possibilita a colaboração entre os utilizadores de uma empresa.

A colaboração que o BiTeamWork permite é ao nível da partilha interna, onde é possível a adição de comentários e anotações aos dados partilhados. Estes comentários são ainda acompanhados de uma pontuação, a qual é dada por outros utilizadores. Quanto mais alta, maior a importância que tem no momento da visualização dos dados. Este *software* permite ainda a adição de segurança na partilha, pelo que os utilizadores podem definir quem pode ver o seu comentário.

- **YellowFin** - A *YellowFin* oferece uma plataforma que permite a análise de dados de diferentes fontes, os quais podem ser visualizados graficamente.

O seu produto permite a partilha interna entre utilizadores. É possível a adição de anotações aos dados partilhados e estes podem ainda ser discutidos através da técnica de conversação referida. Os utilizadores podem ainda seguir um relatório, o que quer dizer que sempre que exista uma atualização, tanto a nível dos dados como dos comentários, estes são notificados.

- **Tibco Spotfire** - A empresa *Tibco* criou o produto *Spotfire*, a qual oferece a descoberta dos dados e a sua visualização. Foca-se muito na área da visualização, oferecendo múltiplos *dashboards*, análises preditivas, análises de localização e também ferramentas de colaboração.

A colaboração no *Tibco* permite uma conversação ao estilo de *chat*, onde os utilizadores envolvidos podem discutir em tempo real vários tipos de dados. Esta conversação é algo similar à existente em plataformas como o *Twitter*⁵ e a *Salesforce*⁶.

- **Birst** - O produto *Birst* oferece a possibilidade de centralizar os seus dados numa plataforma apenas, tendo como origem diferentes conetores, incluindo SAP e *Salesforce*. A interação com os dados é realizada de forma visual e interativa.

O *software* permite uma conversação sobre os dados que estão a ser visualizados. Desta forma, quando os dados são partilhados, é possível a adição de comentários em tempo real e também a adição de anotações.

⁵*Twitter* - <http://www.twitter.com>

⁶*Salesforce* - <http://www.salesforce.com>

- **QlikView** - Plataforma de BI orientada ao utilizador que lhe permite a visualização dos seus dados de negócio, os quais provêm de diferentes fontes de dados.

A colaboração do *QlickView* realça uma questão muito importante, a de segurança. Assim, a partilha é realizada através da geração de um *link* e apenas os utilizadores autorizados conseguem aceder aos dados partilhados.

- **Chartio** - Este sistema permite ao utilizador importar os seus dados de diferentes fontes de dados. A sua visualização é possível, mas após o utilizador configurar os gráficos de acordo com as entidades presentes nos dados. O acesso aos dados pode também ser feito recorrendo a *Structured Query Language* (SQL).

O *Chartio* fornece apenas a possibilidade de partilha por *link*.

- **BIMeAnalytics** - Esta solução oferece a possibilidade de concentrar os dados de uma empresa na *cloud*, na qual se destaca o elevado número de conetores para diferentes fontes de dados que estão disponíveis, podendo a fonte de dados ser online ou *on-premise*.

A partilha disponível no *BIMeAnalytics* é realizada de uma forma automática, onde os seus *dashboards* são entregues aos outros utilizadores diariamente (partilha interna). Uma vez feita esta partilha, qualquer utilizador consegue iniciar uma conversa. É ainda possível a adição de anotações.

- **GoodData** - Esta empresa destaca-se pela sua ferramenta ser baseada na *cloud* e pela possibilidade de análise de uma grande quantidade de dados. Estes dados podem ser posteriormente visualizados graficamente.

O *software* fornece uma interface onde o utilizador consegue partilhar os seus relatórios e gráficos, dando oportunidade aos outros utilizadores de os comentar.

- **Sisence** - A *Sisence* é um produto de BI que inclui um nível de *back-end*, o qual permite aos seus utilizadores juntar e analisar um grande conjunto de dados de diferentes fontes, e um nível de *front-end* que lhe permite a criação de visualizações, como *dashboards* e relatórios.

A colaboração dos seus utilizadores é facilitada através da geração de um *link*, dando acesso aos utilizadores que o recebem de interagir com os dados partilhados.

- **DataRPM** - A *DataRPM* oferece uma solução muito semelhante à *Wizdee*, na medida em que as suas pesquisas são também realizadas em linguagem natural. Os dados podem provir de diferentes fontes de dados, sendo que o sistema modela-os de forma automática. Ao realizar as pesquisas, o utilizador recebe como solução diversos tipos de respostas visuais.

Este produto permite a conversação entre os utilizadores, onde estes conseguem, em tempo real, discutir sobre os dados que estão a ser visualizados.

- **Looker** - O seu sistema permite aos utilizadores conectar a sua plataforma diretamente às suas fontes de dados, sendo que estes nunca são importados para outra base de dados. Esta solução pode ser gerida *on-premise* ou na *cloud* e possibilita a visualização de todos os dados conectados.

Este *software* permite apenas a partilha por *link*.

Análise Comparativa

A tabela 2.1 faz uma correspondência entre as ferramentas de BI referidas e as abordagens de colaboração e partilha que foram descritas anteriormente.

Como pode ser visualizado nesta tabela, funcionalidades como a geração de *link*, partilha interna e conversação são bastante comuns, sendo de referir que algumas têm um nível de complexidade bastante mais elevado que outras. A segurança na partilha é algo que não é especificado na documentação de muitas das ferramentas, pelo que apenas duas referem como lidam com esse aspeto. A partilha através da geração de HTML e a partilha para redes sociais são pouco usadas neste tipo de ferramentas, o que se percebe, pois muitos dos dados devem permanecer restritos aos colaboradores. Uma das funcionalidades referidas foi a partilha para ferramentas de produtividade, que apesar do seu potencial, não é implementada por nenhuma das ferramentas analisadas.

Ferramenta	Geração de Link	Geração de HTML	Partilha para Redes Sociais	Partilha para Ferramentas de Produtividade	Partilha Interna	Conversação	Segurança na Partilha
Panorama	✓	-	-	-	✓	✓	-
ArcPlan	✓	-	-	-	✓	-	-
Tableau	✓	✓	✓	-	-	-	-
BiTeamWork	-	-	--	-	✓	-	✓
YellowFin	-	-	-	-	✓	✓	-
Tibco	-	-	-	-	-	✓	-
Birst	-	-	-	-	✓	✓	-
QlikView	✓	-	-	-	-	-	✓
Chartio	✓	-	-	-	-	-	-
BIMeAnalytics	-	-	-	-	-	✓	-
GoodData	-	-	-	-	✓	-	-
Sisence	✓	-	-	-	-	-	-
DataRPM	-	-	-	-	-	✓	-
Looker	✓	-	-	-	-	-	-

Tabela 2.1: Ferramentas de BI e respetivas funcionalidades.

2.2 Personalização

Com o crescimento da informação existente, existe cada vez mais dificuldade em encontrar a informação que determinados utilizadores procuram. Atualmente, existem várias abordagens capazes de ajudar os utilizadores a lidar com a sobrecarga de informação existente, proporcionando um acesso mais eficiente à informação mais relevante. É neste sentido, que surge a **Personalização**, a qual é responsável pela adaptação da informação apresentada ao utilizador, tendo em conta as diferenças existentes entre os vários utilizadores (Robinson, 2013). Uma aplicação ou *website* sem personalização apresenta o mesmo conteúdo para qualquer utilizador, sendo independente do seu perfil e preferências, oferecendo, deste modo, uma experiência mais limitada. Por outro lado, e recorrendo à personalização, a experiência pode ser consideravelmente melhorada, pois os conteúdos adaptam-se ao utilizador, podendo esta ser implementada das mais variadas formas. Podemos dividir a personalização em duas grandes categorias: a **Personalização Prescritiva** e **Personalização Adaptativa**.

Personalização Prescritiva

Este tipo de personalização é realizada tendo em conta as interações existentes com o utilizador, podendo recorrer a diversas áreas de conhecimento, como é o caso da atividade recente, histórico, comportamento de cliques, contexto, tempo do dia, entre muitas outras. Desta forma, o conteúdo de uma página é adaptada tendo em conta o conteúdo que se aproxime das suas preferências. Este tipo de personalização pode ser ainda dividida em outras duas sub-categorias: Personalização Explícita e Personalização Implícita. A primeira tem como base as preferências introduzidas pelo utilizador, pelo que é da responsabilidade deste a criação de um perfil e o moldar da informação que deseja ser apresentada. A segunda envolve algoritmos mais complexos e é feita de uma forma automática, em que é a própria aplicação ou *website* que monitoriza o comportamento do utilizador na navegação que realiza, permitindo a aprendizagem do comportamento do utilizador, o que permite a apresentação de informação que mais se aproxima do perfil em causa (Robinson, 2013).

Personalização Adaptativa

A personalização adaptativa é um tipo de personalização que tem vindo a ganhar relevo e permite a previsão do conteúdo que o utilizador está a procurar, antes e durante a sua interação. Para poder existir este tipo de previsão, tem que existir uma base de regras pré-definidas, podendo não ser as melhores para o utilizador em questão, visto que estas são idealizadas tendo em conta um bloco de utilizadores com perfis semelhantes. Deste modo, este requisito revela-se como uma das fraquezas da personalização adaptativa, a qual se encontra numa fase pré-matura, mas que tem um grande potencial (Robinson, 2013).

De seguida, irão ser apresentadas algumas das abordagens já seguidas na implementação da personalização, alguns competidores, bem como técnicas exploradas na área de investigação. É importante referir que este é um tema que pode ser muito explorado, pelo que irá ser focada a personalização associada a motores de pesquisa, pois é sobre este tipo de personalização que o estágio recai.

2.2.1 Abordagens

Existem diversas abordagens associadas à implementação da personalização (Santos et al., 2012), mas existem quatro que irão ser realçadas: **Aprendizagem Computacional** (Anand and Mobasher, 2005), **Algoritmos Genéticos** (Mitchell, 1998), **Sistemas de**

Recomendação (Chulyadyo and Leray, 2014) e **Contexto** (Skillen et al., 2012). Em primeiro lugar irá ser dada uma visão geral de cada uma, e de seguida, irão ser referidas algumas das aplicações a motores de pesquisa, as quais foram usadas por diversos investigadores.

Aprendizagem Computacional

A complexidade do algoritmo de um sistema de personalização pode ser bastante elevada (Ling, 2006). De maneira a lidar com esta complexidade, um sistema deve ser dotado da habilidade necessária de aprender sem ser explicitamente programado (Samuel, 1959). É neste campo que surge a aprendizagem computacional - *machine learning* (ML), a qual permite a extração de conhecimento a partir de um conjunto de dados (Mitchell, 1997a). Os algoritmos de ML baseiam-se na criação de modelos, os quais assentam sobre parâmetros pré-definidos (Bishop, 2006), sendo estes alterados à medida que ocorre a aprendizagem. Estes modelos tendem a ser preditivos, podendo prever situações do futuro, ou descritivos, permitindo ao modelo ganhar conhecimento através de dados específicos.

De seguida, irá ser dada uma visão geral das aplicações associadas à aprendizagem máquina, os dados, uma breve referência de alguns dos seus algoritmos e como pode ser feita a sua avaliação.

Aplicação. As aplicações associadas a este tipo de algoritmos é muito vasta, mas podem ser agrupadas em seis categorias (Alpaydin, 2010):

- **Aprendizagem Supervisionada** - o algoritmo gera uma função que consegue mapear os testes de entrada para as soluções desejáveis. Para isso, recebe um conjunto de testes etiquetados e é treinado segundo estes para no futuro conseguir etiquetar novos dados.
- **Aprendizagem Não Supervisionada** - ao contrário da aprendizagem supervisionada, esta não possui testes etiquetados, sendo responsável por criar essas etiquetas. Este tipo de aprendizagem é menos poderosa que a supervisionada, mas pode ser aplicada a um maior número de problemas (Miner et al., 2012).
- **Aprendizagem Semi-Supervisionada** - combina os exemplos etiquetados e não-etiquetados para gerar uma função apropriada.
- **Aprendizagem com Reforço** - algoritmo que através da sequência de boas ações passadas, consegue gerar sequências ainda melhores.
- **Transdução** - semelhante ao algoritmo de aprendizagem supervisionada, mas que se destaca por não gerar uma função, tentando prever as novas soluções baseando-se nos testes de entrada, nos testes de saída e em novos testes.
- **Aprendizagem para Aprender** - baseado em experiência passada, o algoritmo aprende, de uma forma autónoma, segundo vias indutivas.

Dados. Em cima, são referenciadas diversas vezes os dados de entrada e modelos gerados pelos algoritmos de ML. Estes seguem formas específicas, de modo a serem corretamente interpretadas pelos respetivos algoritmos (Witten and Frank, 2005).

- **Dados de Entrada.** Os **dados de entrada** compreendem três partes (Witten and Frank, 2005):

- **Conceito** - conceito a ser aprendido;

- **Instâncias** - representam exemplos independentes do conceito que deve ser aprendido, pelo que o seu conjunto é denominado de conjunto de dados (*dataset*), o qual irá ser usado pelos algoritmos de ML para a aprendizagem;
 - **Atributos (*features*)** - definem as características das instâncias introduzidas através do seu valor.
- **Modelos Gerados.** A representação dos modelos gerados é um fator muito importante, pois a escolha desta irá influenciar o tipo de técnica que irá ser aplicada sobre eles. Assim, as representações mais usadas são as seguintes (Witten and Frank, 2005):
- **Árvores de Decisão** - baseada no algoritmo de *divide-and-conquer*⁷, esta representação usa uma estrutura de árvore. Cada nó interno representa um teste sobre um atributo e cada nível o teste realizado. E, por último, cada folha representa a classe a que os testes realizados deram origem.
 - **Regras de Classificação** - conjunto de regras, em que cada regra possui duas partes: o antecedente (ou pré-condição) e o conseqüente (ou conclusão). O primeiro representa um conjunto de testes aos atributos independentes e a segunda a classe ou a distribuição de probabilidade sobre as classes, as quais são aplicadas sobre as instâncias que a regra engloba. Todos os testes têm que ser bem sucedidos para uma regra ser disparada.
 - **Regras de Associação** - este tipo de representação é muito semelhante ao anterior, exceto no que toca à capacidade de previsão. Estas conseguem prever qualquer atributo ou combinação de atributos, onde não existe uma classe definida.
 - **Árvores para Previsão Numérica** - árvores que podem ser usadas para realizar previsões sobre um atributo alvo do tipo numérico.
 - **Representação baseada em Instância** - ao contrário das representações anteriores, este tipo de representação usa apenas as instâncias diretamente para inferir resultados, o que implica que não existe generalização.
 - **Clusters** - um *cluster* é um grupo de instâncias que são semelhantes entre si. E para isto, existem diversas representações, podendo ir da mais básica em que a cada instância pertence apenas a um número do *cluster*, até a uma mais complexa, em que a mesma instância pode pertencer a diferentes *clusters* com uma probabilidade associada. É ainda possível produzir estruturas hierárquicas em que cada *cluster* pode ser dividido em outros tantos *sub-clusters*.

Algoritmos/Esquemas. De seguida, são referidos alguns dos algoritmos mais usados, sendo focados aqueles que mais importância têm para o trabalho desta tese. Não querendo oferecer ao leitor uma descrição detalhada de cada algoritmo, irá ser dada apenas uma visão geral da ideia de cada um (Witten and Frank, 2005).

⁷Algoritmo de *Divide-and-Conquer* - divide instâncias grandes em menores, e, recursivamente, o algoritmo é aplicado nas instâncias mais pequenas. A solução final é a combinação das soluções de cada sub-instância (Lindenbaum and Samet, 1990)

- **Algoritmos baseados em Árvores de Decisão.** Este tipo de algoritmo representa árvores que permitem classificar diferentes instâncias, ordenando-as segundo os valores de cada atributo. Cada nó neste tipo de árvore representa o atributo segundo o qual se está a classificar, sendo que cada união representa o valor que este pode assumir. No final do algoritmo, e depois de as instâncias terem percorrido toda a árvore, são ordenadas segundo os valores obtidos (Quinlan, 1986).
- **Algoritmos baseados em Regras.** Pensando nas árvores de decisão, consegue-se perceber que o que está implícito é um conjunto de regras, em que cada uma representa cada caminho existente entre a raiz da árvore e um nó (Kotsiantis, 2007). Contudo, estas regras podem ser induzidas a partir de um conjunto de dados de treino, introduzindo-nos ao tipo de **Algoritmos baseados em Regras**. Estes têm como objetivo a construção do menor número conjunto de regras que é consistente com os dados de treino (Aiolli and Sperduti, 2010). Para isto ser possível, existem diversas abordagens que podem ser seguidas. Uma possível é a abordagem por cobertura, em que usando diversas iterações, as regras são identificadas, provocando a divisão das instâncias em diferentes grupos. De seguida, e de maneira a criar uma regra com a maior eficácia possível, são adicionados alguns testes à regra que está a ser construída. O objetivo é a adição do máximo de instâncias possíveis que podem ser englobadas numa classe específica, excluindo o máximo daquelas sem interesse. Existem outras abordagens baseadas no algoritmo de *separate-and-conquer* (Witten and Frank, 2005). Neste caso, a regra é identificada, as instâncias são adicionadas à classe respetiva, as restantes são excluídas e o algoritmo prossegue aplicado apenas a estas últimas. Uma questão muito importante neste tipo de algoritmos é a fiabilidade que é fornecida. Para este problema ser resolvido, é necessário encontrar uma forma que permita avaliar cada regra que é identificada, avaliando o seu erro sobre um conjunto independente de instâncias, as quais se encontram no conjunto de dados de treino.
- **Algoritmos baseados em Redes Bayesianas.** Uma das abordagens estatísticas que é usada em muitos domínios são as redes bayesianas. A estrutura destas é a de um grafo acíclico direto, em que cada nó representa um atributo, onde estes estão ligado de forma direta, evitando ciclos. A base desta abordagem é calcular a distribuição da probabilidade condicionada dos valores do atributo de determinada classe, sabendo os valores dos outros atributos (Witten and Frank, 2005).
- **Algoritmos baseados em Aprendizagem por Instâncias.** Outro algoritmo de ML é o de aprendizagem baseada em instâncias, o qual usa as instâncias de treino para classificar uma nova. Para isso e através de uma função de cálculo de distância pré-definida, determina quais as instâncias mais próximas a esta última e classifica-a com a classe da instância do conjunto de treino que se encontra mais próxima (Witten and Frank, 2005).
- **Algoritmos baseados em Regras Associativas.** Passando a outro tipo de algoritmo, surgem as **Regras Associativas**. Estas são muito semelhantes às regras de classificação, uma vez que o mesmo procedimento é usado para a solução ser gerada, mas destaca-se pois uma regra associativa consegue prever o valor de mais que um atributo. Para esta regra ser criada é necessário um procedimento de indução de regras que avalia a combinação de todos os atributos e de todos os seus valores. Isto resulta num grande conjunto de regras associativas, as quais terão que passar por um processo de triagem segundo a sua precisão e cobertura. O valor da cobertura precisa de ser definido à priori (Witten and Frank, 2005).

- **Algoritmos baseados em *Clustering*.** Outro dos algoritmos mais populares é o *Clustering*, o qual pertence a um tipo de aprendizagem não supervisionada (Miner et al., 2012). O seu objetivo é identificar, de uma forma automática, instâncias que pertencem ao mesmo grupo e agrupá-las em *clusters*. Neste tipo de algoritmos, dois componentes são necessários: um método que permita calcular o nível de similaridade entre instâncias e um outro método para comparar todas as instâncias, sendo que a métrica de semelhança depende do tipo de dados. Outra variável importante é o número de *clusters* que vão ser usados, podendo estes ser definidos à priori ou podem ser calculados internamente através de uma determinada métrica de distância. Esta métrica é frequentemente uma variação do método de distância entre vetores ou então pode ser usada uma métrica de distância numérica, como é o caso da distância Euclidiana.

Os algoritmos de *Clustering* podem ser divididos em três categorias:

- **Hierárquica** - O *clustering* dos grupos é feito iterativamente formando uma hierarquia em cascata do conjunto de *clusters*.
 - **Particional** - Ao contrário do anterior em que a divisão é realizada de uma forma incremental, aqui todos os clusters são produzidos de uma só vez.
 - **Espectral** - Este tipo de algoritmo utiliza operações de matrizes para a redução dimensional e produz os *clusters* com base nessas mesmas reduções dimensionais.
- **Algoritmos de Classificação Numérica.** Muitos dos algoritmos de ML para lidar com atributos numéricos precisam de ser modificados. Ao contrário destes, os algoritmos de classificação numérica não necessitam de tais modificações. Quando o valor de uma classe é numérico, e todos os atributos também o são, uma das técnicas a considerar é a regressão linear. O conceito deste método é atribuir os valores às diferentes classes como uma combinação linear de todos os atributos, em que o peso de cada um é calculado a partir do conjunto de treino. O objetivo é encontrar os melhores pesos que minimizam a soma do quadrado das diferenças entre a classe atual e a predita. Tendo esta base, diversos algoritmos foram aparecendo (Witten and Frank, 2005) como são o caso do *multiresponse linear regression*, *logistic regression* e *perceptron learning rule*. Mas estes são algoritmos simples revelando-se insuficientes para resolver problemas mais complexos. Estes lidam apenas com modelos lineares onde todos os atributos são numéricos. De modo a lidar com esta limitação, surgiram as *Support Vector Machines* (SVM) (Witten and Frank, 2005). Este tipo de algoritmo constrói modelos que são capazes de corresponder novas instâncias a uma determinada classe. Estes modelos são uma representação de pontos no espaço, os quais são divididos por uma linha. Os pontos que pertencem a um lado da linha são pertencentes a uma determinada classe e os outros, pertencentes ao outro lado da linha, são de outra classe. Apenas a título de curiosidade, os algoritmos de SVM, para além de lidarem com classificação linear, podem também tratar de classificação não-linear, usando para isso o chamado truque de *kernel*, o qual permite encontrar e estudar os tipos de relações existentes (por exemplo classificações, correlações, pontuações) em *datasets*.
- **Algoritmos baseados em Redes Neurais.** Este tipo de algoritmos são inspirados no cérebro humano, os quais conseguem encontrar estruturas de dados e algoritmos para o processo de aprendizagem e classificação de dados (Freeman and Skapura, 1991). O modelo mais simples de uma rede neuronal pode ser visualizado

na figura 2.2. Aos dados de entrada introduzidos ($X_0 - X_n$) é atribuído um conjunto de pesos ($W_0 - W_n$), podendo estes variar entre 0 e 1. Se a soma destes ultrapassar o limite (θ), um sinal é despoletado para a saída. Caso contrário, nenhum sinal é enviado. A aprendizagem deste tipo de algoritmos é baseada no ajustamento dos valores atribuídos aos pesos de modo que o modelo gerado seja o desejado, usando para isso uma aprendizagem supervisionada. Neste tipo de técnica de aprendizagem, os padrões são conhecidos à priori e o conjunto de dados de treino estão classificados com as etiquetas desejadas. Antes de o algoritmo ser iniciado, os pesos são inicializados com valores aleatórios e cada conjunto de treino é associado a uma rede neuronal. Quando o processo é despoletado, cada conjunto de dados de entrada gera um modelo, sendo este comparado com o modelo desejado. Se for o modelo correto, os valores dos pesos não são alterados, caso contrário, esses valores devem ser ajustados de maneira a conduzir a solução para a desejada.

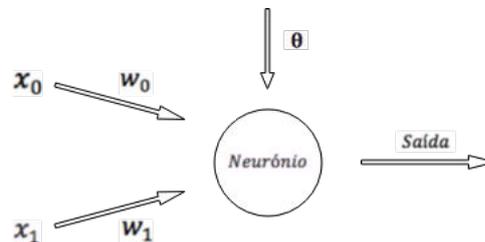


Figura 2.2: Ilustração da rede neuronal mais simples.

- **Algoritmos Evolucionários.** Algoritmos baseados na evolução das espécies (Mitchell, 1998). Neste trabalho irão ser referidos mais à frente em detalhe os algoritmos genéticos, os quais são um tipo de algoritmos evolucionários. Estes merecem uma posição de destaque devido à sua importância na realização deste trabalho.

Avaliação. Por último, e depois de o algoritmo gerar as soluções para o problema em questão, procede-se à avaliação do modelo criado, constituindo um passo muito importante para a validação do algoritmo usado. Este processo é dividido em várias partes diferentes, pelo que vão ser apenas referidas algumas das mais importantes (Witten and Frank, 2005)

- **Treino e Testes** - O conjunto de dados usado para a validação de um algoritmo é algo que tem que ser tido em conta. Por exemplo, se os testes forem realizados sobre o conjunto de dados de treino, o erro vai ser muito menor do que aquele que vai ser obtido quando aplicado a um novo conjunto de dados. Por essa razão, é necessária a existência de dois conjuntos de dados, os de treino e os de teste. A existência de um terceiro conjunto depende do método de aprendizagem usado. Alguns métodos usam o conjunto de treino para construir uma estrutura básica e usam um outro conjunto, denominado de conjunto de validação, para otimizar os parâmetros dessa mesma estrutura. Para serem obtidos resultados mais fiáveis, todos os conjuntos referidos devem ser diferentes. Uma outra variável a ter em conta é o número de instâncias presentes no conjunto de testes, uma vez que uma percentagem de sucesso associada a um conjunto maior tem mais valor do que quando associada a um conjunto pequeno.
- **Estimativa de Erro** - Para o cálculo da estimativa de erro de determinado algoritmo de ML ser possível, é necessária uma correta escolha das instâncias que são usadas para treino e para teste. Neste sentido, existem diversos métodos que permitem realizar esta divisão. Primeiramente, pode ser referido o *holdout*, que consiste

na divisão do conjunto de dados em que uma parte vai para treino e outra para teste. Neste método pode ser logo identificado uma grande limitação se imaginarmos que os dados que determinam uma determinada classe ficam apenas representados no conjunto de teste. De forma a ultrapassar esta limitação, este método pode ser enriquecido com uma técnica chamada de estratificação, que garante que em ambos os conjuntos referidos existem instâncias que representem todas as classes. Este tipo de métodos em que é usada a mesma informação tanto para treino como para testes tem sempre as suas limitações. De forma a contornar isto, surge o método de *cross-validation*, que permite treinar o classificador com todos os dados, mas mantendo sempre conjuntos de treino e testes diferentes. Assim, este método permite escolher um número fixo de partições em que é feita a divisão do conjunto de informação e o processo de treino é aplicada sobre $(n-1)/n$ partições, em que n representa o número fixo referido, deixando de fora uma das partições para teste. Este só termina quando todas as partições forem tomadas como teste, sendo que no final é feita uma média da estimativa de erro calculada para cada uma. Este é um dos métodos estatísticos mais utilizados, mas simples variações são possíveis, como por exemplo a aplicação da técnica já referida de estratificação.

- **Estimativa de Custo** - O custo dos classificadores tomarem decisões erradas é um fator que também deve ser tido em conta. Para isso, associada à variável de estimação de erro devem ser tidas em conta outras variáveis como o custo despendido no processo de predição ou o custo despendido no processo de treino.
- **Avaliação da Predição Numérica** - Neste tipo de avaliação, os princípios de medição usados na classificação também podem ser usados, mas tem que ser feita uma alteração, a medida de qualidade associada à estimação do erro. Isto, porque ao contrário do que acontece na classificação em que o erro é simplesmente binário, o valor deste pode vir em diferentes números.

Algoritmos Genéticos

Os algoritmos genéticos (AG) são também algoritmos de aprendizagem computacional, pertencendo ao tipo de algoritmos evolucionários. Serão descritos separadamente devido à sua grande relevância para a realização desta tese.

Os AG são algoritmos de pesquisa heurística adaptativa, baseados no princípio darwiniano de seleção natural (Mitchell, 1998). Estes permitem a resolução de problemas de otimização e pesquisa, através de uma forma inteligente de exploração aplicada a uma pesquisa aleatória. Apesar do carisma aleatório, estes algoritmos permitem o direcionamento da pesquisa para a região onde existe maior probabilidade de encontrar a melhor solução.

As técnicas usadas nos AG simulam o processo natural de evolução, no qual existe a sobrevivência dos mais aptos ao longo das diferentes gerações para a resolução de um problema. Cada geração é composta por um conjunto de indivíduos (população), onde cada qual é representado por um conjunto de genes, formando os cromossomas. Cada indivíduo representa um ponto no espaço de pesquisa e uma solução possível. A modificação e a interação dos diferentes indivíduos é que contribuem para o processo evolutivo. Esta analogia com a estrutura genética e comportamento dos cromossomas está assente sobre os seguintes fundamentos:

- Os indivíduos numa população competem por recursos e companheiros.
- Os indivíduos mais aptos produzem uma melhor descendência do que os indivíduos menos aptos.

- Os melhores genes são transmitidos de geração em geração, havendo casos em que os progenitores produzem descendentes com melhores genes do que eles próprios.
- A cada sucessiva geração, a população encontra-se mais adaptada ao ambiente.

Processo Evolutivo. O processo usado nos AG, representado na figura 2.3, é iniciado pela criação de uma população aleatória (Cunha et al., 2012). Neste processo iterativo, ou seja, a cada geração, ocorre uma seleção dos indivíduos promissores, sobre os quais vão ser aplicadas operações genéticas de cruzamento e mutação para a criação dos seus descendentes e, por fim, a inserção destes novos indivíduos na população, marcando desta forma uma nova geração. Neste processo não existe a garantia de progresso nem que é encontrada a solução ótima, mas esta dinâmica tende que ao longo das gerações estejam a ser produzidas soluções candidatas cada vez mais adaptadas.

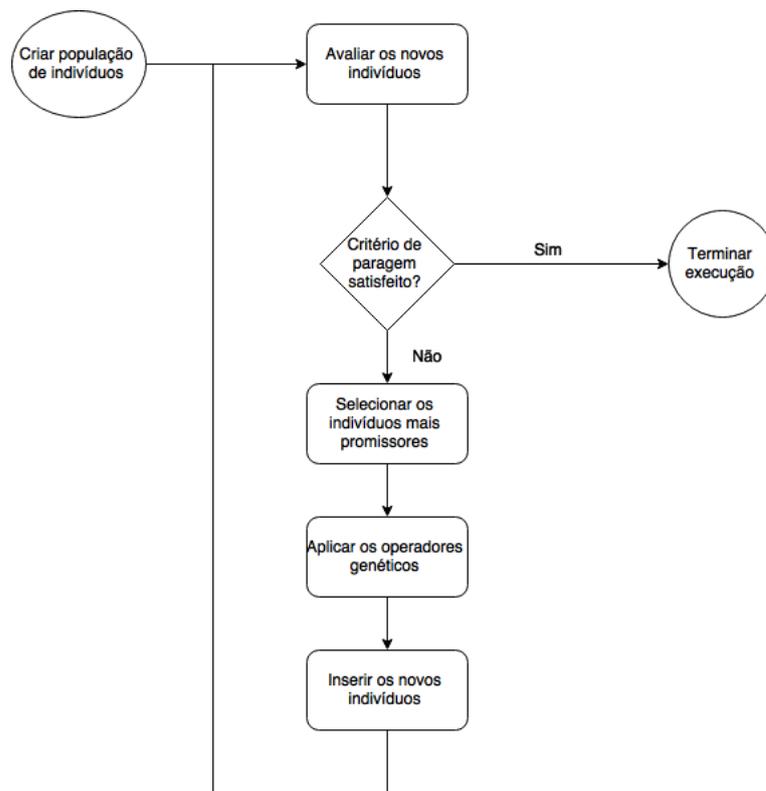


Figura 2.3: Processo evolutivo usado nos algoritmos genéticos (Cunha et al., 2012).

As etapas do processo evolutivo referidas acima são sintetizadas por:

- Criação da população - Estipula como os primeiros indivíduos são criados. Aqui é importante definir a forma de representação destes (cromossomas), pois vai ser sobre esta que as operações hão-de ser aplicadas.
- Avaliação dos indivíduos - Função que calcula o valor de *fitness*⁸ de um indivíduo, sendo este que deve ser maximizado.
- Critério de paragem - Determina quando o processo deve ser interrompido. Normalmente, são usados os seguintes critérios:

⁸*Fitness* - Aptidão de um indivíduo, a qual é feita recorrendo a uma função de avaliação (Mitchell, 1998).

- Tempo limite de execução.
 - Número máximo de gerações.
 - Valor de *fitness* atingido.
 - Estagnação do processo.
- Seleção - Define como é realizada a escolha dos indivíduos mais aptos em detrimento dos menos aptos. A diferença que existe entre a probabilidade de serem escolhidos os melhores indivíduos e a probabilidade de serem escolhidos os piores é denominado de pressão de seleção. Quando esta é alta, apenas os indivíduos com melhor *fitness* são escolhidos, em média. Por outro lado, quando é baixa, os piores indivíduos têm mais probabilidade de serem selecionados. Apesar de parecer vantajoso, a escolha de uma pressão natural alta pode levar a um problema de convergência prematura, pois não existe diversidade suficiente na população, sendo que as melhores soluções podem estar noutras zonas do espaço de pesquisa que já não são novamente atingidas. Alguns dos métodos mais usados na seleção são os seguintes:
 - Roleta. Imagine-se uma roleta do casino em que cada parcela (indivíduo) tem uma dimensão proporcional ao seu valor de *fitness* (Cunha et al., 2012). De seguida, o indivíduo é escolhido aleatoriamente, sendo semelhante ao processo da roleta quando gira (2.4).

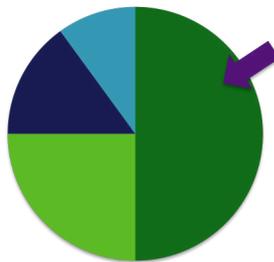


Figura 2.4: Seleção por roleta.

- Amostragem Universal Estocástica. Muito idêntica ao método da roleta, diferenciando-se na medida em que apenas o primeiro indivíduo é escolhido aleatoriamente. A partir desse ponto, os restantes indivíduos são escolhidos avançando em passos igualmente desfasados à volta da roleta. O tamanho destes passos é determinado pelo número de indivíduos a serem selecionados.
- Torneio. São escolhidos dois indivíduos aleatoriamente e através de um torneio é escolhido um ou outro. Este torneio baseia-se num valor de probabilidade que é gerado entre 0 e 1, o qual é comparado com um valor de probabilidade de seleção pré-determinado. Assim, se esse valor for igual ou inferior ao valor pré-determinado, então o indivíduo mais apto é escolhido. Se for superior, o indivíduo escolhido é o mais fraco.
- Elitismo. Este método surge na medida em que bons candidatos podem ser perdidos ao longo do processo evolutivo. Para contrariar isto, o elitismo copia uma pequena proporção dos melhores candidatos para a geração seguinte. Os restantes são escolhidos aleatoriamente.

- Operadores Genéticos - Operadores que definem como são criados os novos indivíduos. Existem dois tipos:

- Cruzamento - Combina o material genético de dois indivíduos, de modo a ser criado um novo descendente. Os métodos mais comuns neste tipo de operação são os seguintes:

- * Cruzamento por Um Ponto. É escolhida aleatoriamente uma posição de uma lista de dois indivíduos e um descendente é gerado. Este possui a lista do primeiro progenitor desde o início até ao ponto escolhido, sendo o resto da lista completada com a do segundo progenitor desde o ponto até ao fim. Uma ilustração deste processo é representado na figura 2.5.



Figura 2.5: Cruzamento por um ponto.

- * Cruzamento por Dois Pontos - Muito parecido com o cruzamento por um ponto, sendo diferente uma vez que são escolhidos dois pontos em vez de apenas um. Desta maneira, o descendente irá ter a representação do primeiro progenitor até à primeira posição, juntamente com a representação do segundo progenitor desde a primeira posição até à segunda, terminando com a representação do primeiro progenitor desde o segundo ponto até ao final.
- * Cruzamento Uniforme - Aleatoriamente é escolhido material dos dois descendentes que quando juntos formam um novo indivíduo.
- * Cruzamento Aritmético - Se a representação dos indivíduos for binária, o cruzamento aritmético pode ser aplicado. Este permite realizar operações aritméticas entre os progenitores para criar um descendente. Exemplos deste tipo de operações são o AND e o OR.
- Mutação - Permite criar um novo indivíduo a partir de outro. Estas alterações permitem criar pequenas variações num indivíduo, de modo a ser possível a exploração das suas zonas vizinhas. Dois tipos de mutação são listados de seguida:
 - * Inversão de *Bits* - Se o indivíduo tem uma representação binária, então alguns dos seus *bits* são invertidos.
 - * Mudança de Ordem - Neste tipo de mutação existe uma alteração nas posições dos valores de representação do indivíduo.
- Inserção - Determina a forma como os novos descendentes são inseridos na população. Existem duas formas (Syswerda, 1989; Whitley, 1989):
 - Geracional - Os novos indivíduos são adicionados a uma lista temporária e quando esta atinge o tamanho da população inicial substitui a população principal.

- *Steady-State* - Os novos indivíduos ao serem criados são logo inseridos na população principal, sendo que determinados indivíduos têm que ser eliminados de modo a manter o tamanho inicial da população.

Avaliação. Para a validação deste tipo de algoritmos, o processo é muito semelhante ao usado na avaliação referida na aprendizagem máquina.

Sistemas de Recomendação

Um sistema de recomendação (SR) pode ser definido como um sistema informático que tem como objetivo recomendar, para uma coleção de utilizadores, um conjunto de itens que podem ser do seu interesse (Melville and Sindhvani, 2010). O processo de recomendação tem como base estimar a utilidade de um elemento específico para cada utilizador. Esta utilidade é geralmente representada por uma pontuação, sendo um elemento mais útil para um utilizador quando maior for essa pontuação. Esta pode ser explicitamente dada pelo utilizador, mas a inexistência desta revela-se como um problema para os sistemas de recomendação, os quais têm que realizar estimativas para obter o valor de cada par utilizador/elemento. Baseando-se nas estimativas obtidas, os sistemas de recomendação ordenam os seus resultados, podendo oferecer ao utilizador um conjunto de elementos úteis, ou oferecer um conjunto de utilizadores para um determinado elemento. Alternativamente, alguns sistemas estimam apenas as preferências relativas do utilizador, conhecida como filtragem baseada em preferência (Cohen et al., 1999).

O cálculo da estimativa referida é um dos maiores problemas neste tipo de sistemas, pelo que o seu cálculo pode seguir diferentes abordagens, das quais se destacam a colaborativa, a baseada em conteúdo, a baseada em conhecimento e a híbrida (Jannach et al., 2010).

Recomendação Colaborativa. O conceito base da recomendação colaborativa é explorar a informação sobre as opiniões de um conjunto de utilizadores e recomendar um conjunto de elementos em que o utilizador atual está provavelmente interessado (Melville and Sindhvani, 2010).

Uma das abordagens seguidas neste tipo de técnica é a recomendação baseada no vizinho mais próximo do utilizador. Os vizinhos mais próximos são aqueles que no passado tiveram preferências semelhantes ao utilizador atual. Desta maneira, a utilidade de um elemento, que é desconhecido para o utilizador atual, é calculada a partir da pontuação dada por esses vizinhos. Aqui existem as assunções de que se os utilizadores tiveram gostos semelhantes no passado, então gostos semelhantes ocorrerão no futuro, e, também, que as preferências dos utilizadores mantêm-se constantes ao longo do tempo.

Um dos problemas encontrados na abordagem referida é o peso computacional que esta tem num sistema onde existam milhões de utilizadores. Torna-se impossível realizar recomendações em tempo real. Neste seguimento, surge outra abordagem, a recomendação baseada no vizinho mais próximo do elemento. Sendo esta baseada nos elementos, pode ser pré-processada em modo *offline*, tornando possível a recomendação em tempo real, mesmo para uma grande quantidade de dados. Ao contrário da anterior, a ideia principal deste método é prever a utilidade usando a semelhança entre os elementos. Assim, se diferentes utilizadores considerarem úteis dois elementos, então um outro utilizador que tenha a mesma opinião num deles, também considerará útil o outro, sendo este valor uma média entre as pontuações que foram dadas a esse elemento pelos outros utilizadores.

Contudo, esta segunda abordagem não consegue resolver dois problemas que ambas as abordagens possuem: o problema de *cold-start* (Schein et al., 2002) e o *sparsity* da pontuação (Papagelis et al., 2005). O primeiro refere-se ao facto de terem que existir um número

mínimo de elementos com pontuações pré-determinadas para o sistema de recomendação conseguir fornecer recomendações úteis. O outro problema prende-se com o número de elementos com pontuação ser muito inferior ao número de pontuações que têm que ser estimadas, levando a uma fraca performance do sistema.

Recomendação baseada em Conteúdo. Seguindo uma abordagem baseada em conteúdo, a recomendação sugerida são elementos que são semelhantes em conteúdo a outros elementos que o utilizador já considerou úteis no passado ou são correspondentes aos atributos do utilizador (Jannach et al., 2010). Para isto, dois requisitos têm que ser cumpridos: tem que existir uma descrição do elemento e tem que existir um perfil de utilizador, capaz de fornecer informação acerca dos seus interesses passados (Melville and Sindhvani, 2010).

Um elemento é, normalmente, caracterizado por um perfil, o qual é construído a partir da extração das suas características (Basu et al., 1998). Esta extração pode ser realizada construindo modelos a partir dos dados, usando para isso aprendizagem estatística e aprendizagem computacional (Mitchell, 1997b).

Neste tipo de recomendação, um problema é a dependência da qualidade e quantidade das características associadas aos elementos. Outro problema é a sobre-especialização, tendo como consequência o sistema recomendar sempre elementos semelhantes aqueles já conhecidos pelo utilizador.

Recomendação baseada em Conhecimento. Este tipo de recomendação é baseado no conhecimento explícito fornecido pela caracterização do elemento, pelas preferências do utilizador e pelo critério de recomendação (Jannach et al., 2010). Imaginemos a escolha de um novo computador. Um sistema de recomendação que usa as abordagens anteriores, usa pontuações de elementos com anos de vida. Num cenário destes não faz sentido, pelo que o uso de recomendação baseada em conhecimento torna-se inevitável (Melville and Sindhvani, 2010). Este resolve problemas de *cold-start*, uma vez que não necessita de pontuações pré-definidas.

Podem ser considerados dois tipos de recomendação baseada em conhecimento: baseada em limitações (Felfernig and Burke, 2008) e baseada em casos (McSherry, 2005). A primeira procura elementos que correspondem às regras de recomendação que foram introduzidas. Por outro lado, a segunda usa métricas de semelhança entre os elementos para obter elementos semelhantes, os quais obedecem aos requerimentos especificados pelo utilizador.

Recomendação Híbrida. As abordagens anteriormente referidas têm os seus prós e contras, como os problemas de *cold-start* ou *data sparsity* (Jannach et al., 2010). Nenhuma tem o potencial necessário para conseguir tirar o máximo partido das recomendações fornecidas. Para isso, a construção de um sistema híbrido tornou-se objeto de estudo de muitos investigadores. Este sistema tenta combinar as forças dos diferentes algoritmos e modelos, de maneira a tornar o sistema de recomendação mais poderoso, tentando assim ultrapassar alguns dos problemas já referidos.

Avaliação. Para a validação deste tipo de algoritmos, o processo é muito semelhante ao usado na avaliação referida na aprendizagem máquina.

Considerações Finais. As aplicações a motores de pesquisa deste tipo de técnicas é algo que tem sido muito explorado, de modo a adaptar a informação ao utilizador em questão. Seja na construção da pesquisa, como resultado de *autocomplete*, seja na ordenação dos resultados que são exibidos, esta técnica é algo que é usado, tendo como base alguns

dos estudos que são feitos sobre o utilizador. Estes estudos podem ser feitos através do perfil que o utilizador definiu, através dos seus *cookies*, ou mesmo através do seu histórico e sequência de ações que está a executar. Nas secção seguinte, onde são descritos trabalhos relacionados, vão ser referidas algumas destas técnicas com mais detalhe.

Contexto

O diálogo entre seres humanos é algo natural em que os envolventes conseguem trocar ideias com sucesso e conseguem reagir adequadamente ao que é dito, pelo menos em grande parte das vezes. Isto acontece devido à linguagem que partilham, mas muito se deve à perceção daquilo que os rodeia. Esta perceção, este conhecimento implícito é fruto do contexto.

No entanto, este contexto, e apesar de ser algo natural entre os seres humanos, é bastante difícil de ser transferido para o mundo computacional. Os computadores não se encontram preparados para usar o potencial do contexto na sua interação com os utilizadores. Para fazer uso deste no domínio informático, Dey and Abowd (Dey and Abowd, 2000) define-o da seguinte forma:

"Contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, local, ou objeto que é considerado relevante para a interação entre um utilizador e uma aplicação, incluindo o próprio utilizador e a própria aplicação."

Zimmerman et al. (Zimmermann et al., 2007) consegue ainda dividir os elementos que podem ser usados na descrição do contexto em cinco categorias:

- Individualidade - Informação que pode ser relacionada com a entidade em questão;
- Atividade - Informação das atividades que o utilizador está a realizar e pretende atingir;
- Localização - Importante em aparelhos portáteis;
- Tempo - Informação está muito relacionada com a dimensão temporal;
- Relações - Relação existente entre uma entidade e as restantes.

Tendo uma ideia mais clara do que é o contexto, podem-se perceber as vantagens que este é capaz de oferecer. Muitas das abordagens em sistemas de recomendação não fazem uso do contexto, desperdiçando o seu enorme potencial. Contudo, começam a aparecer algumas que tiram partido desta informação.

O contexto pode ser integrado em sistemas de recomendação, sendo usado para influenciar os resultados obtidos pelo sistema, em conjunto com os interesses do utilizador e informação dos elementos (Adomavicius and Tuzhilin, 2011). Para esta integração ser possível, o contexto pode ser definido como um conjunto de atributos pré-definidos, os quais não são alterados significativamente ao longo do tempo (Dourish, 2004), ou podem ser modelizados interativamente e guardados numa memória de curto prazo (Anand and Mobasher, 2007). Esta informação contextual pode ser obtida de três formas:

- Explícita - Obtida diretamente das pessoas ou fontes de informação relevantes;
- Implícita - Obtida a partir dos dados ou do ambiente;
- Inferida - Inferido através de técnicas de estatística ou *data mining* (Witten and Frank, 2005).

Assim, os sistemas de recomendação podem usufruir do contexto seguindo duas vias: procura e pesquisa orientada ao contexto, a qual restringe a pesquisa a um conjunto de informação e apenas retira as mais relevantes, ou licitação e estimativa da preferência contextual, usando técnicas de modelização e aprendizagem sobre as preferências do utilizador, tendo que para isso monitorizar as suas interações ou obter a partir da sua experiência (Adomavicius and Tuzhilin, 2011).

2.2.2 Trabalhos Relacionados

O tema da personalização é um dos muitos abordados por investigadores em todo o mundo. De seguida, serão mencionadas algumas das abordagens já seguidas quando aplicadas a motores de pesquisa, bem como os resultados alcançados.

A personalização de motores de pesquisa já foi alvo de muitas abordagens diferentes. As primeiras baseavam-se nos interesses que eram introduzidos pelos utilizadores. Estes interesses eram usados para a filtragem dos resultados obtidos através da semelhança entre o conteúdo presente nas páginas *web* e os interesses do utilizador (Pretschner and Gauch, 1999). No entanto, estas abordagens revelaram-se com pouco potencial, uma vez que os estudos apontam que a maioria dos utilizadores sentem-se reticentes em providenciar explicitamente os seus interesses e a sua opinião sobre os resultados obtidos (Carroll and Rosson, 1987).

Os estudos posteriores em personalização de motores de pesquisa focam-se na aprendizagem automática das preferências dos utilizadores através das suas ações e não pelo que estes oferecem (Shen et al., 2005b). rae Kim and Chan (2005) propuseram a aprendizagem do perfil de utilizador, *User Interest Hierarchy* (UIH), através dos seus interesses, baseando-se para isso nas páginas web presentes nos favoritos de cada um e no algoritmo de *Divisive Hierarchy Clustering* (DHC) (Kim and Chan, 2003). Este algoritmo particiona, recursivamente, as palavras em *clusters* cada vez mais pequenos até a condição de paragem ser atingida e calcula a força de relacionamento entre os termos originados, sendo esta força tanto maior quando ambas as palavras são encontradas na mesma página web, presente nos favoritos do utilizador. Desta maneira, a função de pontuação é aplicada sobre os resultados que foram retornados na pesquisa e são ordenados segunda essa pontuação. Na figura 2.6 é exibida uma visão geral do processo descrito.

A performance deste tipo de métodos de *clustering* baseia-se em parâmetros como o número de *clusters* e as suas formas (Sheikh et al., 2008). No entanto, na maioria dos casos da vida real o número de *clusters* é desconhecido. Nenhum destes algoritmos de *clustering* são capazes de formar grupos naturais com base em todos os padrões de entrada de uma forma eficiente e automática, especialmente quando o conjunto de dados tende a ser grande. Este problema, em muitos dos casos, deve-se à escolha incorreta dos centros dos *clusters* iniciais. Um *cluster* otimizado pode ser obtido se forem usados algoritmos evolucionários como os algoritmos genéticos. Estes são capazes de melhorar a performance de algoritmos *k-means* em algumas tarefas, podendo ainda otimizar o número de *clusters* em tarefas que possuam *clusters* bem formados e separados.

Liu et al. (2002) propõem que um conjunto de categorias seja atribuída a cada utilizador, através da análise do seu histórico. Posteriormente, as pesquisas são então mapeadas segundo esse conjunto, servindo como um meio de desambiguação das palavras introduzidas na pesquisa.

Noutro trabalho, uma nova técnica é apresentada, propondo combinar dois módulos: o perfil do utilizador e o seu padrão de navegação (Liu et al., 2010). Quando um utilizador realiza uma pesquisa, a extração dos termos-chave é realizada e esses termos são comple-

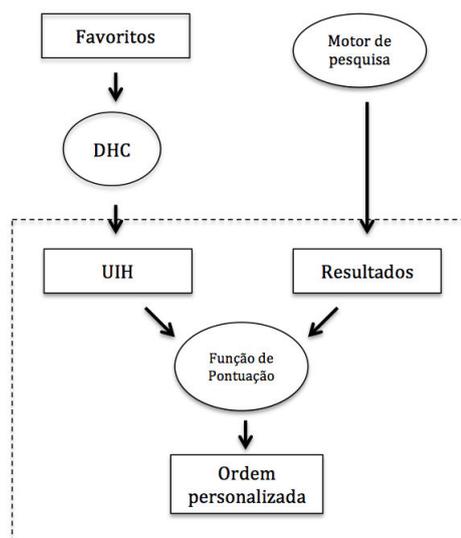


Figura 2.6: Personalização de resultados a partir dos favoritos (rae Kim and Chan, 2005).

mentados com pesos. Este termo e peso são comparados com as categorias existentes na base de dados, de forma a encontrar a intenção do utilizador. Por outro lado, sobre o padrão de navegação do utilizador é aplicado um algoritmo de classificação Bayesiano, o qual é capaz de gerar um padrão. O resultado dos módulos referidos é então mapeado de modo a ser possível encontrar os dados que vão de encontro às pretensões do utilizador. Assim, usando o primeiro módulo é possível encontrar as intenções do utilizador e, usando o segundo, é possível encontrar os dados que são frequentemente utilizados. Posteriormente, os resultados dos módulos são comparados de maneira a conseguir o máximo de interseções, gerando uma página *web* personalizada. Este processo pode ser visualizado na figura 2.7.

Outra abordagem usada na personalização é a previsão das referências futuras (Cadez et al., 2000), baseando-se para isso no padrão de navegação extraído do histórico da sessão. Os autores usam um modelo de Markov⁹ para aprender e representar as dependências entre as referências das páginas e assim prever para onde a pesquisa deve ser apontada.

Um dos problemas que preocupa a comunidade científica é a diversificação de resultados de uma pesquisa. Liang et al. (2014) propõem uma estratégia de aprendizagem supervisionada, a qual se baseia na implementação de *Dirichlet topic model* sobre os documentos de interesse do utilizador. Esta implementação permite decompor uma coleção de documentos em tópicos, representando cada documento como um sub-conjunto destes tópicos. Após a extração de tópicos ser realizada, estes são cruzados com aqueles extraídos dos documentos devolvidos pela pesquisa, de maneira a torná-los consistentes com os interesses do utilizador.

Para evitar a ambiguidade foi feita a introdução do contexto nos motores de pesquisa (Teevan et al., 2011). Assim, são implementadas técnicas de personalização baseada em memória, as quais possibilitam a aprendizagem das associações diretas entre os pares pesquisa-URL (dada uma pesquisa, o utilizador escolhe constantemente aquele URL em particular). Shen et al. (2005a) desenvolveram um modelo de linguagem baseado em

⁹Modelo de Markov - modelo estocástico usado para modelar sistemas que estão constantemente em mudança, onde é assumido que os estados futuros apenas dependem do estado presente e não da sequência de eventos que o precedem (Rabiner, 1989)

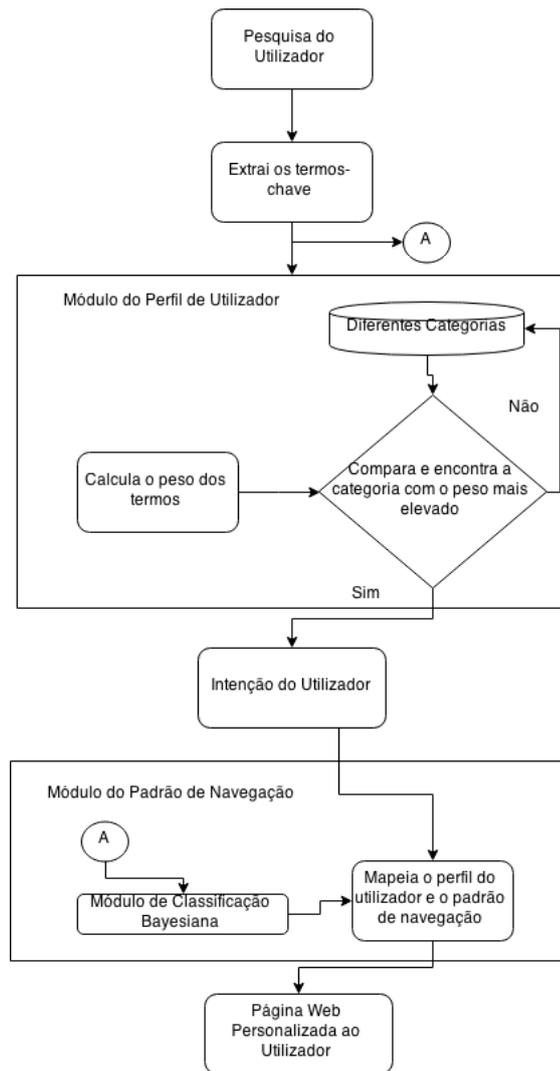


Figura 2.7: Personalização de resultados a partir dos módulos de perfil de utilizador e do seu padrão de navegação (Liu et al., 2010).

contexto, introduzindo o histórico do utilizador e as pesquisas precedentes para uma personalização de curto prazo. Contudo, este tipo de modelos é apenas específico para o contexto atual e não podem ser bem generalizados para lidar com as preferências gerais do utilizador.

Nas abordagens referidas, apenas a informação proveniente do utilizador ou das suas ações é usada para criar o seu perfil. Existem também estratégias que incorporam as preferências de grupos de utilizadores na personalização dos resultados de uma pesquisa (Sugiyama et al., 2004; Sun et al., 2005). Para isso, a recomendação colaborativa é o método típico para personalização baseada em grupos (Sun et al., 2005). (Sugiyama et al., 2004) propuseram um novo método *CubeSVD*¹⁰ capaz de correlacionar utilizadores, pesquisas e páginas *web* disponíveis no histórico. Este método permite, através dos dados do histórico de diferentes utilizadores, criar associações entre as três dimensões referidas, sendo gerado

¹⁰*CubeSVD* - Abordagem baseada no método *Singular Value Decomposition* (SVD), o qual permite a fatorização de uma matriz real ou complexa, tendo o prefixo *Cube*, pois assenta em três dimensões - utilizadores, pesquisas e páginas *web* (Sugiyama et al., 2004).

o seguinte quarteto $\langle u, q, p, w \rangle$, onde w representa o interesse do utilizador na página p quando u introduz a pesquisa q . Assim, as páginas *web* podem ser recomendadas a u de acordo com os pesos associados ao par $\langle u, q \rangle$.

Na implementação deste tipo de técnicas existe a preocupação da mudança de perfil do utilizador (Masurel and Lefèvre-Hasegawa, 2014). Quando os interesses do utilizador são constantes o suficiente, então a atividade passada a longo prazo é adequada para a personalização da sua pesquisa (Teevan et al., 2005b). Por outro lado, se o utilizador tem uma súbita mudança de interesses, a atividade referida não deve ser usada, dando preferência à atividade mais recente - curto prazo (Chirita et al., 2006). É necessário então encontrar soluções para identificar este tipo de mudanças. (Eickhoff et al., 2013). Neste sentido, os autores propõem modelar o perfil do utilizador segundo os seus interesses, preferências e estilo de pesquisas do passado e, depois, calculam a divergência que existe entre esse perfil e as novas pesquisas e interesses do utilizador. Esta divergência pode ser calculada a partir de duas categorias: termos usados na pesquisa e tópicos. Na primeira é recolhida, para cada utilizador, a frequência com que é usado cada termo da pesquisa durante a modelação do perfil. Esta contagem é feita também em sessões mais recentes. Desta forma, é possível a projeção destas contagens num espaço-vetor. A métrica a ser usada é a distância que existe entre os termos anteriormente usados e os mais recentes. Para os tópicos, o mesmo processo é usado, sendo a métrica a distância que existe entre os tópicos anteriormente pesquisados e os atuais. Nestas medidas, quanto maior a distância, maior a probabilidade de os interesses do utilizador terem alterado.

Alguns investigadores chegaram à conclusão que o processo de personalização tem mais eficiência numa pesquisa do que noutras (Teevan et al., 2005a). Para pesquisas menos ambíguas, os resultados normais de um motor de pesquisa são suficientes, levando a que a personalização seja desnecessária. Neste sentido, as pesquisas podem ser divididas em: pesquisas claras, pesquisas semi-ambíguas e pesquisas ambíguas (Chirita et al., 2006, 2005). Os autores concluíram que a personalização aumenta significativamente a qualidade dos resultados para pesquisas semi-ambíguas e ambíguas, mas para pesquisas claras, sem ambiguidade, é preferível o uso de motores de pesquisa comuns, sem personalização. Outros autores conseguem ainda dividir as pesquisas em pesquisas novas e pesquisas recorrentes (Tan et al., 2006). Baseando-se apenas no histórico mais recente, os resultados para as pesquisas novas tornaram-se mais interessantes quando comparado com o uso do histórico completo. Ao contrário destas, as pesquisas recorrentes devolveram resultados mais eficientes quando é usado todo o histórico. Estes resultados levam a concluir que nem todas as pesquisas devem ser personalizadas da mesma forma.

Capítulo 3

Abordagem

O presente capítulo apresenta a abordagem. Irá ser iniciado com um enquadramento do *Wizdee Discovery*, onde irá ser dada a descrição de algumas funcionalidades do sistema, apresentando, de seguida, os objetivos do estágio, tanto a nível de colaboração como de personalização. A análise de requisitos irá ser outra das das secções abordadas, bem como a arquitetura da solução proposta. Tendo uma visão da estrutura interna da empresa e objetivos do autor, são apresentadas os componentes relacionados com a implementação das novas funcionalidades. No final, é apresentada uma análise de riscos.

3.1 Enquadramento

O produto *Wizdee Discovery* engloba um conjunto de funcionalidades e termos próprios. Esta secção pretende dar a conhecer alguma dessa terminologia, a qual vai ser usada nas secções seguintes.

Como já foi referido anteriormente, o *Wizdee Discovery*, sendo um produto de BI, possibilita a geração de relatórios baseados em dados de negócio. Estes, denominados de livros no sistema, estão organizados em páginas e *widgets*. Desta maneira, um livro pode conter várias páginas, e uma página pode ser composta por vários *widgets*, como pode ser observado na figura 3.1. Estes últimos podem ser tabelas ou gráficos representantes dos dados pesquisados. Os *widgets* podem ainda ser agregados em *dashboards*, os quais têm o intuito de mostrar a informação que o utilizador acede com maior frequência de uma maneira mais facilitada.

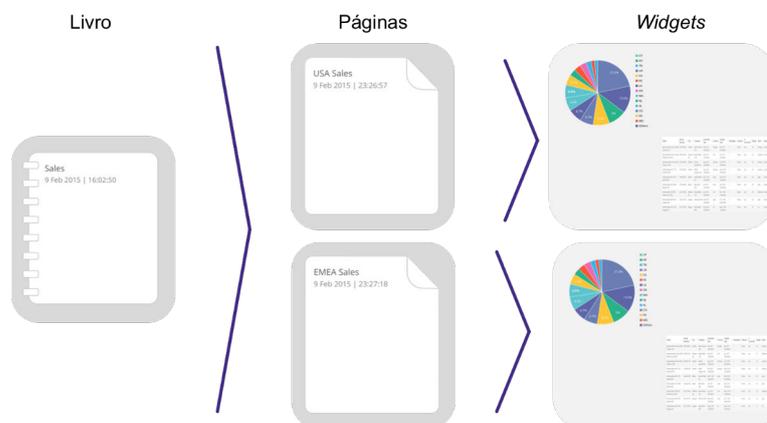


Figura 3.1: Organização dos livros.

Como esta ferramenta permite a visualização de dados de negócio, muitas vezes inte-

ressa a partilha destes entre diferentes utilizadores. Atualmente, a plataforma possibilita a partilha interna de relatórios/livros entre grupos. Um grupo é um conjunto de utilizadores. Após a partilha ser realizada, todos os utilizadores envolvidos têm acesso ao conteúdo partilhado, tendo estes apenas permissão de leitura.

Na figura 3.2, pode ser visualizada a barra de pesquisa inicial do sistema. É através desta que o *Wizdee Discovery* pretende ser inovador, possibilitando ao utilizador a procura dos dados que deseja através de uma pesquisa em linguagem natural, ou seja, linguagem humana. Para auxiliar este processo, existem três funcionalidades que se destacam, sendo o caso das **Pesquisas Sugeridas**, do **Autocomplete** e dos **Cenários**.

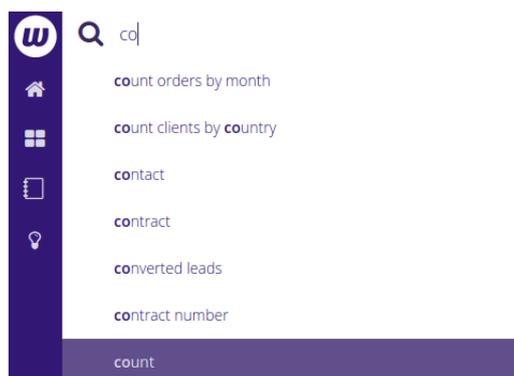


Figura 3.2: Barra de pesquisa com *autocomplete*.

Pesquisas Sugeridas

As pesquisas sugeridas englobam três tipos de pesquisa, as pesquisas que são introduzidas à priori no sistema, as que vão sendo usadas frequentemente pelo utilizador, bem como as marcadas como favoritas. Estas são exibidas no *autocomplete*, o qual irá ser especificado de seguida.

Autocomplete

De maneira a guiar o utilizador na procura dos dados, o sistema é dotado da funcionalidade de *autocomplete*. Este tem como intuito mostrar um leque de possibilidades de pesquisas a que a plataforma é capaz de responder, evitando que o utilizador realize pedidos que não têm uma resposta válida e, ao mesmo tempo, ajudar o utilizador a construir a sua pesquisa. Para isso, à medida que é introduzida uma letra, a lista de *autocomplete* que é exibida vai sendo moldada, apresentando apenas resultados que contêm o que foi escrito na barra de pesquisa. O utilizador pode, desta maneira, navegar pelas possibilidades que vão sendo mostradas, procurando facilitar a procura dos dados que pretende. O *autocomplete* é também relevante quando o utilizador não sabe qual o tipo de dados e informação que irá encontrar no sistema. Com esta funcionalidade isso já não acontece, uma vez que o *autocomplete* lhe vai mostrando as possibilidades que pode utilizar. Neste leque são ainda exibidas as pesquisas sugeridas, as quais foram descritas anteriormente. Na figura 3.2 pode ser visualizado este processo a decorrer.

Cenários

Uma das maiores preocupações no processamento da linguagem natural é a ambiguidade que as pesquisas podem conter. Para o sistema processar o pedido do utilizador, a pesquisa

introduzida em linguagem natural deve ser convertido em linguagem perceptível pelo sistema, sendo que esta pode ter diferentes interpretações, uma vez que os mesmos termos podem existir em diferentes tipos de dados. As interpretações referidas são chamadas, no âmbito do *Wizdee Discovery*, de cenários. Dependendo da pesquisa introduzida, diferentes cenários são gerados. Estes são ordenados segundo algoritmos de evolução genética e aprendizagem máquina, exibindo ao utilizador, o cenário com maior probabilidade de ser o correto. Caso este seja mal interpretado, o utilizador tem ainda a hipótese de poder alterar para outro, do leque de possibilidades existente.

De seguida é apresentado um exemplo de uma listagem de cenários/interpretações feita pelo sistema à pesquisa "*Accounts by date*":

1. *Accounts by Created Date of Lead*
2. *Accounts by Converted Date of Lead*
3. *Accounts by Closed Date of Opportunity*
4. *Accounts by Created Date of Opportunity*

3.2 Objetivos

Nesta secção irão ser apresentados os objetivos principais do estágio, os quais podem ser divididos em três áreas: colaboração (3.2.1), ordenação de cenários (3.2.2) e personalização (3.2.3).

3.2.1 Colaboração

Um dos aspetos importantes na análise de dados de negócio é a habilidade para colaborar e partilhar informação. O sistema da *Wizdee* fornece ao utilizador funcionalidades colaborativas capazes de transmitir e discutir os relatórios analíticos com outros utilizadores. Na abordagem anterior existia apenas a possibilidade de partilha interna de relatórios entre grupos, mas um dos objetivos passou por alargar este leque, podendo assim existir a partilha de páginas, *widgets* e *dashboards*, tanto internamente como externamente, entre utilizadores e grupos. Para isso, foi implementado a partilha por email, bem como a partilha para ferramentas de produtividade, ou seja, ferramentas que são utilizadas por um grande número de utilizadores na organização da sua vida pessoal e trabalho, sendo exemplo destas, a Dropbox, Evernote¹ ou GoogleCalendar². Ainda no campo da colaboração, outro dos objetivos passou pela possibilidade de adição de comentários nos relatórios, páginas, *widgets* e *dashboards*, com suporte para menções a outros utilizadores, outros relatórios ou outros *dashboard*. Para os utilizadores poderem saber que algo foi partilhado com eles, se existem comentários novos num relatório ou *dashboard* seu ou se foram mencionados num comentário, foi também implementado o suporte para notificações. Desta forma, sempre que existe um novo evento relacionado com o utilizador, este é prontamente notificado.

Uma visão geral dos objetivos da colaboração podem ser visualizados na figura 3.3. Aqui o utilizador que aparece no centro partilha um relatório para um utilizador e para um grupo, sendo estes prontamente notificados. Partilha também dois *widgets* para as redes sociais e para as suas ferramentas de produtividade.

¹Evernote - <https://www.evernote.com>

²Google Calendar - <https://www.google.com/calendar>



Figura 3.3: Visão geral dos objetivos da colaboração.

3.2.2 Ordenação de Cenários

A otimização da ordenação de cenários apresenta uma parte fulcral na realização deste trabalho, pois é sobre esta que a implementação da personalização assenta.

O primeiro objetivo passou por o estagiário ambientar-se à abordagem já usada pela empresa anteriormente e que trouxe bons resultados. Essa abordagem baseia-se no uso de um algoritmo genético que é usado para calcular os pesos inerentes a diferentes *features*. Esta abordagem permite atribuir uma pontuação correta a cada cenário e, desta forma, ordenar os cenários de acordo com a sua relevância.

Numa primeira fase, foi necessário adaptar o uso do algoritmo genético a uma nova versão do motor de pesquisa do *Wizdee Discovery*, pois o AG estava adaptado a uma versão antiga do mesmo. Este novo motor foi implementado por outro elemento da equipa de desenvolvimento que não o estagiário e foi necessário pois o anterior encontrava-se com uma estrutura que era difícil de sofrer modificações. Então a solução encontrado foi fazer uma reestruturação desse código e limpar coisas que eram desnecessárias e acrescentar outras que faziam falta. Com estas alterações, o algoritmo genético ficou incompatível com esta versão do motor, levando a que fosse necessário a alteração das várias ligações e dependências.

Numa segunda fase para a implementação desta abordagem foi necessária a identificação de mais *features* que pudessem ser usadas para melhorar a ordenação de cenários.

O segundo objetivo, e depois de obtidos os valores dos pesos para as diferentes *features*, passou pela implementação destas no motor de pesquisa. Dependendo do peso calculado pelo AG, cada uma tem a sua influência e a interação das diferentes *features* calcula a pontuação final para cada cenário, melhorando desta forma a ordenação de cenários.

3.2.3 Personalização

Numa perspectiva de personalização, o sistema adequa a informação que apresenta consoante o utilizador que está a explorar o sistema, pelo que a informação referida não é apenas baseada na relevância de cada termo introduzido, mas também no histórico do utilizador em questão, tendo em conta métricas como as pesquisas realizadas com maior frequência, as opções que toma ao longo da visualização dos seus dados, sendo o caso da alteração de cenário, e também do contexto apresentado. Nesta linha, a ferramenta adequa o que é apresentado às necessidades de cada um, permitindo o acesso a informação relevante com maior facilidade.

Para tornar isto possível, a moldagem do contexto do utilizador e a aprendizagem do seu comportamento na exploração dos seus dados foi necessária, pelo que o objetivo foi melhorar as funcionalidades já existentes de *autocomplete*, ordenação de cenários e pesquisas sugeridas. A primeira é agora capaz de sugerir as pesquisas que o utilizador poderá estar interessado em realizar. Na segunda, o cenário exibido é personalizado, de modo a permitir a escolha da melhor interpretação possível à pesquisa realizada, tendo como base o contexto apresentado. O último objetivo passou por melhorar o processo de sugestões de pesquisa que o utilizador realiza de seguida, tendo em conta as que foram realizadas anteriormente.

Uma visão geral dos objetivos da personalização podem ser visualizados na figura 3.4. Aqui é possível ver que através das interações do utilizador com o sistema é modelizado o seu perfil e este é usado para a ordenação de cenários, nas sugestões de *autocomplete* e também nas pesquisas sugeridas, tornando os resultados exibidos adaptados ao utilizador.

3.3 Análise de Requisitos e Restrições

A análise de requisitos e restrições é um passo muito importante na implementação de qualquer projeto. Esta permite um bom planeamento do que irá ser realizado, possibilitando uma melhor visão sobre o produto final. Nesta secção irão ser referidos os requisitos funcionais, bem como os requisitos não-funcionais, englobando os requisitos de desempenho, qualidade e linguísticos. Em termos de restrições irão ser referidas as restrições tecnológicas.

3.3.1 Requisitos Funcionais

Os requisitos funcionais são aqueles que definem as funções que foram realizadas, descrevendo o conjunto de entradas, comportamentos e saídas. Esta secção irá ser dividida nas duas áreas principais deste trabalho: colaboração e personalização.

Colaboração

A tabela 3.1 descreve os requisitos funcionais respeitantes à área de colaboração, os quais foram pensados tendo por base a análise de competidores e idealizando onde o estagiário poderia ser diferenciador. Estes requisitos foram validados recorrendo a uma série de testes unitários.

RF.01 - Partilha Interna. A partilha interna permite aos utilizadores partilhar e trocar relatórios analíticos e *dashboards* sem recorrer a uma ferramenta externa. Tudo é partilhado no seio da plataforma, pelo que um utilizador pode partilhar os seus livros, páginas, *widgets* e/ou *dashboards* com outros utilizadores e/ou grupos. Para isso, basta escolher o utilizador/grupo desejado ao carregar no botão de partilha que é exibido junto

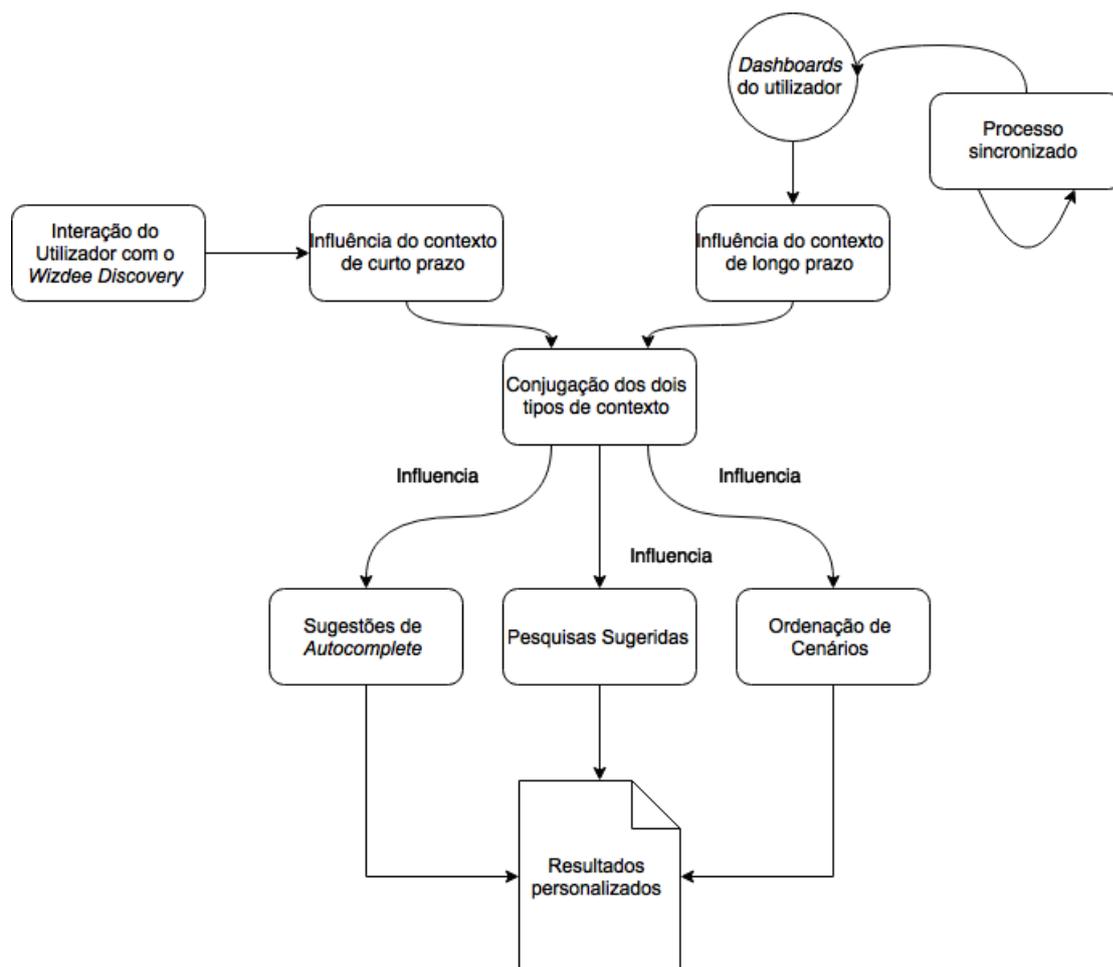


Figura 3.4: Visão geral dos objetivos da personalização.

ID	Nome	Descrição	Dependências
RF.01	Partilha Interna	Os utilizadores do sistema devem poder partilhar entre si os relatórios que estão a realizar, bem como os <i>dashboards</i>	—
RF.02	Partilha Externa	O sistema deve permitir ao utilizador partilhar os relatórios e os <i>dashboards</i> que desejar através de diferentes canais.	—
RF.03	Suporte para Comentários	O sistema deve possibilitar ao utilizador a adição de comentários sobre os relatórios e <i>dashboards</i> pretendidos.	RF.01;RF.02
RF.04	Suporte para Notificações	O sistema deve facultar ao utilizador notificações quando algo é partilhado com ele.	RF.01;RF.02;RF.03

Tabela 3.1: Requisitos Funcionais para Colaboração.

do livro ou seus descendentes e *dashboard*. A partir deste momento, todos os utilizadores com quem foi realizada a partilha têm acesso à informação, tendo estes permissão só de leitura.

RF.02 - Partilha Externa. A partilha externa, ao contrário da interna, permite a partilha de dados de negócio através de canais de comunicação externos. Para isso, esta pode ocorrer pela partilha de um *link* gerado pelo sistema, por email, por redes sociais ou para uma ferramenta de produtividade.

Deste modo, por *link*, é possível a partilha de relatórios completos ou apenas de uma página ou *widget*, sendo possível a definição das permissões dadas, podendo ser de leitura ou escrita. É ainda possível a atribuição de um tempo de validade, findo o qual o acesso ao *link* e aos dados partilhados torna-se impossível.

Na partilha por email, todo o tipo de dados pode ser partilhado, pelo que basta ao utilizador introduzir os emails dos destinatários, o assunto e o texto. Os dados partilhados são enviados como anexo.

As redes sociais escolhidas para a partilha são o *Facebook*, *LinkedIn* e *Twitter*. Para isto, o utilizador efetua a autenticação, aceita que a aplicação tenha autorização para aceder ao seu perfil e pode partilhar os dados que pretende, acompanhados do texto que desejar.

À semelhança do que ocorre com as redes sociais, em que o utilizador tem que se autenticar e dar permissões à aplicação para aceder à sua conta, a partilha para ferramentas de produtividade tem que seguir os mesmos passos. Desta maneira, o utilizador pode adicionar os seus relatórios, respetivos constituintes e *dashboards* à *Dropbox*. No *Evernote* tem a possibilidade de adicionar notas a um determinado livro de notas escolhido pelo utilizador. Outra das possibilidades é a marcação de um evento no *Google Calendar*, podendo adicionar descrições e convidar outros intervenientes. Por último, o utilizador pode ainda adicionar uma nova tarefa no *Trello*.

Na tabela 3.2 pode ser visto um resumo do que foi descrito, onde são representados os requisitos funcionais da partilha externa.

ID	Nome	Descrição	Dependências
RF.02.1	Partilha por <i>link</i>	Os utilizadores devem poder partilhar os seus dados através de um <i>link</i> gerado pelo sistema.	—
RF.02.2	Partilha por <i>Facebook</i> ³	Os utilizadores do sistema devem poder partilhar os <i>widgets</i> no seu perfil do <i>Facebook</i> .	—
RF.02.3	Partilha por <i>Twitter</i>	Os utilizadores do sistema devem poder partilhar os <i>widgets</i> no seu perfil do <i>Twitter</i> .	—
RF.02.4	Partilha por <i>LinkedIn</i> ⁴	Os utilizadores do sistema devem poder partilhar os <i>widgets</i> no <i>LinkedIn</i> .	—
RF.02.5	Adicionar Nota no <i>Evernote</i>	Os utilizadores do sistema devem poder adicionar uma nota na sua conta do <i>Evernote</i> .	—
RF.02.6	Adicionar Ficheiro na <i>Dropbox</i>	Os utilizadores do sistema devem poder adicionar o seu livro, página, <i>widget</i> ou <i>dashboard</i> na sua conta <i>Dropbox</i> .	—
RF.02.7	Marcar Evento no <i>Google Calendar</i>	Os utilizadores do sistema devem poder marcar um evento no seu <i>Google Calendar</i> .	—
RF.02.8	Adicionar Tarefa no <i>Trello</i> ⁵	Os utilizadores do sistema devem poder adicionar uma tarefa na sua conta do <i>Trello</i> .	—

Tabela 3.2: Requisitos Funcionais da Partilha Externa.

RF.03 - Suporte para Comentários. O sistema deve permitir a adição de comentários, de modo a possibilitar o utilizador a colocar diversas notas associadas aos dados pretendidos. Este pode ainda fazer referências a outros utilizadores, outros relatórios ou *dashboards* através de menções. Para isso basta introduzir um "@", caso seja um utilizador, ou "#", caso seja um livro, página, *widget* ou *dashboard*.

RF.04 - Suporte para Notificações. O utilizador deve receber notificações sempre que algo seja partilhado com ele, que sejam feitos comentários num relatório seu ou que este seja mencionado num comentário. Assim, terá acesso a uma zona de atividade onde

poderá ver estas interações.

Personalização

Analisada a primeira área, surge a tabela 3.3, referente aos requisitos funcionais da parte de personalização, os quais surgiram depois da análise de alguns trabalhos relacionados e como outros motores de pesquisa lidavam com a ambiguidade nas suas pesquisas. Estes requisitos foram validados usando testes quantitativos, onde foram comparadas a abordagem atual e a abordagem com as novas funcionalidades em algumas métricas e, também, usando testes qualitativos, os quais refletiram a opinião de utilizadores no uso da abordagem personalizada.

ID	Nome	Descrição	Dependências
RF.05	Definição do Contexto do Utilizador	O sistema deve ser capaz de definir o contexto do utilizador.	—
RF.06	Personalização das Sugestões de <i>Autocomplete</i>	O <i>autocomplete</i> deve ser personalizado para cada utilizador, sendo divulgadas as pesquisas que mais se aproximam do seu perfil, e tendo em conta o contexto.	RF.05
RF.07	Personalização da Ordenação dos Cenários	O sistema deve ser capaz de escolher o cenário mais apropriado para cada utilizador, segundo as suas opções anteriores e segundo o contexto.	RF.05
RF.08	Pesquisas Sugeridas	O sistema deve ser capaz de sugerir uma pesquisa que seja do interesse do utilizador. Na impossibilidade de serem apresentados resultados a determinada pesquisa, uma outra é sugerida.	RF.05

Tabela 3.3: Requisitos Funcionais para Personalização.

RF.05 - Definição do Contexto do Utilizador. Com base na interação do utilizador com o produto, o sistema deve ser treinado de maneira a conseguir moldar o contexto do utilizador. Por exemplo, imaginando um gestor que é responsável por analisar de forma sistemática as vendas da sua empresa, bem como as suas evoluções. O seu contexto deve ser moldado de forma a que os resultados que lhe são exibidos estejam relacionados com estas vendas, prevenindo ambiguidade nas suas pesquisas.

De salientar que o contexto vai ser melhorado ao longo da experiência do utilizador, uma vez que o algoritmo de aprendizagem vai tendo mais dados.

RF.06 - Personalização das Sugestões de *Autocomplete*. Voltando ao exemplo dado em cima, em que existe um gestor responsável por analisar as vendas da sua empresa. Nesta situação, o *autocomplete* deve sugerir, de uma forma eficaz, uma pesquisa relacionada com as vendas.

De salientar que esta funcionalidade vai sendo melhorada ao longo da experiência do utilizador.

O contexto também irá influenciar as sugestões apresentadas.

RF.07 - Personalização da Ordenação dos Cenários. À semelhança do que ocorre na implementação do RF.06, esta funcionalidade também tem em consideração o contexto na escolha dos cenários e respetiva ordenação. Recorrendo a um novo exemplo, consideremos a situação que um utilizador ao realizar determinada pesquisa procede sempre à alteração do cenário, pois o que é exibido não é o pretendido. Neste caso, o sistema deve

aprender as preferências do utilizador e exibir o cenário correto nas próximas vezes que realizar a mesma pesquisa ou relacionadas com esta. Dependendo do contexto, a mesma pesquisa poderá também devolver diferentes resultados.

RF.08 - Pesquisas Sugeridas. Realizando o reconhecimento do contexto de utilizador, o sistema deve sugerir uma pesquisa do interesse do utilizador.

Quando o utilizador introduz uma pesquisa e não é possível a apresentação de resultados, uma nova pesquisa é sugerida, na qual existe a garantia da obtenção de resultados, tendo como base duas métricas: nível de semelhança com a pesquisa anteriormente realizada e contexto.

3.3.2 Requisitos de Desempenho

Cenário de Caso de Uso: Um utilizador ao usar a abordagem atual e a abordagem com as novas funcionalidades não deteta uma diferença de mais que um segundo nos tempos de resposta.

- **Teste.** Para validação serão comparados o tempo de resposta entre a abordagem atual e a abordagem com as novas funcionalidades, verificando se existe alguma diferença entre os tempos superior a um segundo.

3.3.3 Requisitos de Qualidade

Cenário de Caso de Uso: Dois utilizadores distintos utilizam o *Wizdee Discovery* e o contexto moldado é independente de um e de outro.

- **Teste.** Serão realizados mini-questionários a diferentes utilizadores para averiguar se o contexto está a ser bem aplicado.

Cenário de Caso de Uso: Um utilizador realiza um conjunto qualquer de pesquisas e verifica que o número de respostas corretas é melhorada com a introdução das novas funcionalidades.

- **Teste.** Para validação será comparado o número de respostas corretas dadas pela abordagem atual e pela abordagem com as novas funcionalidades através de um conjunto de pesquisas.

Cenário de Caso de Uso: Um utilizador consegue realizar qualquer pesquisa na abordagem atual e consegue realizar a mesma na abordagem com contexto.

- **Teste.** Para validação será comparada a posição do melhor cenário seguindo a abordagem atual e posição desse mesmo cenário usando a abordagem com as novas funcionalidades. Se a posição não for igual a -1, então as duas abordagens conseguem responder à pesquisa introduzida pelo utilizador.

3.3.4 Requisitos Linguísticos

A tabela 3.4 expõe os requisitos linguísticos.

ID	Nome	Descrição	Dependências
RL.01	Língua Portuguesa	As pesquisas poderão ser realizadas em Português.	—
RL.02	Língua Inglesa	As pesquisas poderão ser realizadas em Inglês.	—

Tabela 3.4: Requisitos Linguísticos.

ID	Nome	Descrição	Dependências
RT.01	Linguagem de Programação	Todas as funcionalidades deverão ser implementadas em Java.	—

Tabela 3.5: Restrições Tecnológicas.

3.3.5 Restrições Tecnológicas

As restrições tecnológicas são apresentadas na tabela 3.5.

A única restrição apresentada deve-se ao *Wizdee Discovery* estar desenvolvido em Java.

3.4 Arquitetura

A presente secção pretende dar uma visão geral da arquitetura da plataforma que vai ser usada. Esta segue um modelo que assenta em três diferentes camadas: a **camada de dados** (3.4.1), a **camada de negócio** (3.4.2) e a **camada de apresentação** (3.4.3). A primeira é responsável pela persistência dos dados, enquanto que a segunda pela manipulação destes. Por último, é necessário que estes dados sejam apresentados ao utilizador, tendo a camada de apresentação a responsabilidade desta tarefa. A figura 3.5 apresenta a arquitetura do sistema, onde as setas representam o fluxo de controlo.

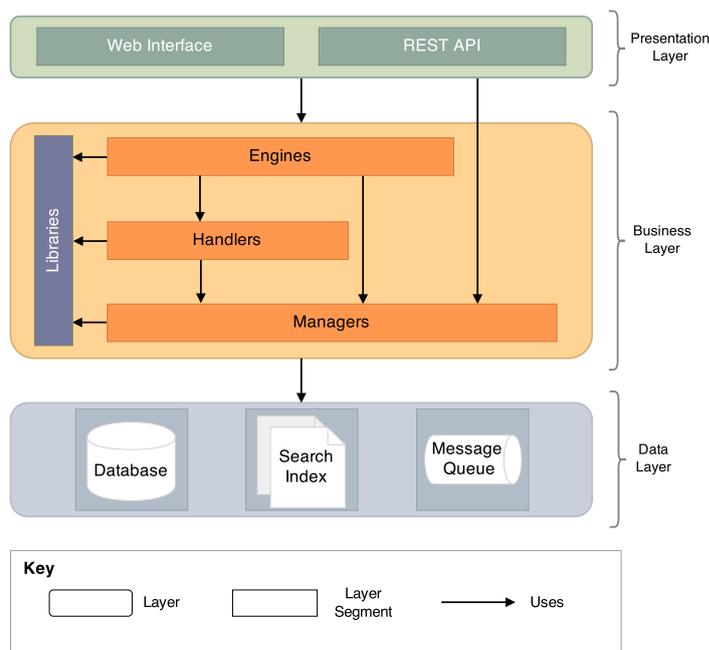


Figura 3.5: Arquitetura do sistema.

3.4.1 Camada de Dados

A camada de dados é responsável pela persistência dos dados indispensáveis ao correto funcionamento do sistema. Este armazena as estruturas necessárias para a operação do sistema, bem como para a otimização de tarefas, como a pesquisa e a representação das relações entre os dados. Os componentes que constituem esta camada são os seguintes: índice de pesquisa, base de dados e fila de mensagens.

Índice de Pesquisa

A pesquisa de texto numa base de dados relacional é uma tarefa pesada, pelo que o índice de pesquisa facilita essa tarefa. Com base na plataforma de pesquisa empresarial Apache Solr⁶, a qual é implementada sobre o motor de pesquisa textual Apache Lucene⁷, os dados são indexados e as pesquisas sobre estes são bastante eficientes.

Base de Dados

Na base de dados é onde estão guardados os dados indispensáveis para o funcionamento do sistema. Atualmente, a base de dados usada é o *PostgreSQL*⁸, sendo este um sistema de gestão de dados objeto-relacional desenvolvido como *open-source*.

Fila de Mensagens

A fila de mensagens fornece um serviço de mensagens assíncrono que possibilita aos produtores colocar tarefas na fila, sendo estas armazenadas até serem consumidas. Posteriormente, as tarefas são consumidas e executadas. As mensagens são guardadas na fila de forma persistente, de forma a que na eventualidade de o sistema ficar indisponível, as mensagens não se percam e o processo possa recomeçar a partir do ponto onde foi interrompido.

De momento, a fila de mensagens está a ser usada no sistema para sincronização de dados de fontes de dados externas. Quando é realizado o pedido para a importação dos dados de uma base de dados remota, estes são divididos em vários pedidos mais pequenos. Estes representam as tarefas que são colocadas na fila, as quais são posteriormente consumidas, sendo os dados armazenados na base de dados local e indexados no índice de pesquisa.

3.4.2 Camada de Negócio

Na camada de negócio existem três tipos de componentes, os quais interagem entre si para uma correta manipulação dos dados. Estes são os seguintes: *Managers*, *Handlers* e *Engines*.

Managers

Os *managers* realizam as operações básicas sobre os dados presentes na camada de dados. Deste modo, conseguem oferecer acesso aos dados abstratamente, permitindo aos outros componentes este acesso de uma forma transparente.

⁶Apache Solr - <http://lucene.apache.org/solr/>

⁷Lucene - <http://lucene.apache.org/>

⁸PostgreSQL - <http://www.postgresql.org>

Handlers

Estes componentes têm a responsabilidade de processar o texto de um pedido e devolver uma resposta. Como cada *handler* pode gerar uma resposta, cada pedido poderá ter várias. Neste caso, será o *engine* que invocou os *handlers* que terá a função de escolher qual a resposta ou respostas a serem apresentadas.

Engines

Como foi dito acima, os *engines* têm a função de coordenar as respostas vinda dos *handlers* para determinado pedido, funcionando como um agrupador destes. Contêm os métodos necessários para a escolha da resposta ou respostas mais adequadas, para a gestão da sessão e para a gravação das estatísticas relacionadas com a geração da resposta. Estes dotam o sistema de funcionalidades de alto nível, permitindo o suporte à pesquisa.

3.4.3 Camada de Apresentação

A camada responsável pela interação entre o utilizador e o sistema é a camada de apresentação. Nesta estão inseridas as aplicações Web que permitem a interação do utilizador com o sistema. De modo a viabilizar a comunicação destas com as restantes camadas, estão ainda inseridos nesta camada os *endpoints* da API, implementados usando uma abordagem REST, os quais fornecem as funcionalidades indispensáveis para a integração completa do sistema.

3.5 Componentes

Na secção anterior foi dada uma visão geral da arquitetura do sistema, onde na camada de negócio foram descritos os vários tipos de componentes existentes. Os componentes que vão ser referidos de seguida são aqueles sobre os quais o trabalho foi desenvolvido, começando por apresentar os componentes que foram alterados, aqueles que foram criados, passando, posteriormente, para os componentes que foram usados.

3.5.1 Componentes Alterados

Os componentes alterados são aqueles que precisaram de sofrer remodelações para serem capazes de implementar os requisitos referidos.

Query Manager

O *query manager* é um dos fornecedores de informação do *autocomplete manager*, o qual tem a responsabilidade de dar sugestões de pesquisa ao utilizador. Este tem a função de obter as pesquisas sugeridas, bem como as mais frequentes, e provisionar o *autocomplete manager* com estas.

Este componente foi alterado de maneira a permitir que as pesquisas dadas por este fossem corretamente ordenadas segundo o contexto do utilizador.

Schema Manager

À semelhança do anterior, este componente é mais um fornecedor de informação do *autocomplete manager*, tratando da informação relativa aos esquema de dados, tendo em conta as tabelas e colunas existentes.

Os dados fornecidos por este componente tiveram que ser mais uma vez ordenados segundo o contexto do utilizador.

Workspace Manager

O *workspace manager* é o componente que trata de gestão dos relatórios, permitindo a adição de livros, páginas, *widgets* e *dashboards*, bem como, a sua edição, remoção e obtenção.

É sobre este que foram implementados os métodos relativos à colaboração, alterando alguns dos já existentes e criando novos. Métodos de gestão dos relatórios e *dashboards* foram modificados de maneira a ser possível a adição de informação relativa à partilha interna de dados entre diferentes utilizadores. Para a partilha externa, novos métodos foram criados, os quais permitiram o envio da informação desejada para determinado destinatário, e, também, a conexão com algumas das ferramentas de produtividade que o utilizador possua conta.

Description Handler

O *description handler* possui os métodos necessários para a geração e ordenamento dos cenários de acordo com o pedido que é realizado.

Desta maneira, para permitir a personalização dos resultados obtidos, o método de ordenamento foi alterado, de modo a possibilitar a influência do contexto do utilizador na escolha do cenário.

Retrieval Handler

O *retrieval handler* possui os métodos necessários para a geração de uma resposta através de um determinado cenário.

Este componente foi importante na medida em que possibilitou a criação de eventos importantes para a implementação final.

3.5.2 Componentes Criados

Os componentes criados são aqueles que ainda não existem e precisam de ser criados para a implementação das novas funcionalidades. Inclui-se neste grupo o componente *Context Manager*.

Context Manager

O *context manager* foi um dos componentes que teve que ser implementado de raiz e tratou da personalização, a qual é constituída por dois tipos de modelos: modelo persistente e modelo temporário.

O primeiro modelo referido irá persistir o contexto de longo prazo do utilizador na base de dados, o qual foi modelado através dos *dashboards* gravados pelo utilizador.

Por outro lado, o modelo temporário não persiste os dados, sendo estes perdidos quando o utilizador sai do sistema. Este modelo tem em conta o contexto a curto prazo, o qual é modelado com base nas interações que tem com o sistema bem como as últimas pesquisas realizadas.

3.5.3 Componentes Usados

Os componentes usados são aqueles em que o seu acesso é feito apenas pela API e são usados pelos componentes referidos anteriormente. Não foram necessárias quaisquer alterações nos métodos existentes.

Autocomplete Manager

Este componente tem a responsabilidade de obter as várias opções para serem exibidas ao utilizador no *autocomplete*. Para isto, estas opções são provenientes de outros *managers* que se registam como prestadores de informação, tendo o *autocomplete manager* a função de organizar este conjunto de informação de acordo com a relevância de cada uma. Exemplos deste são o *query-manager* e *schema-manager*, os quais já foram referidos anteriormente.

Este *manager* apenas tem a função de agregar as sugestões de pesquisa dadas pelos outros *managers*.

3.6 Análise de Riscos

Qualquer projeto está sujeito a riscos que podem comprometer o seu desenvolvimento, pelo que é importante haver uma prévia reflexão sobre estes, antecipando problemas que possam ocorrer e avaliando fluxos de integração. Desta forma, nesta secção são apresentados os riscos que foram previstos, bem como o respetivo plano de mitigação que poderia minimizar os efeitos negativos desses contratemplos.

Na tabela 3.6 podem ser vistos os vários riscos que foram identificados para a realização deste trabalho. Já na figura 3.6 estes riscos estão dispostos numa matriz de exposição, a qual relaciona a probabilidade de ocorrência com o impacto que poderá trazer ao desenvolvimento.

ID	Nome
Rs.01	Incapacidade para desenvolver algoritmo com desempenho adequado
Rs.02	Tempo despendido na execução de treino e testes
Rs.03	Restrições do software herdado
Rs.04	Dependência de terceiros
Rs.05	Atraso no desenvolvimento de uma tarefa
Rs.06	Pouca experiência do estagiário no desenvolvimento de algoritmos genéticos
Rs.07	Impossibilidade de testar em ambiente real

Tabela 3.6: Lista de Riscos e respetivo Grau de Risco.

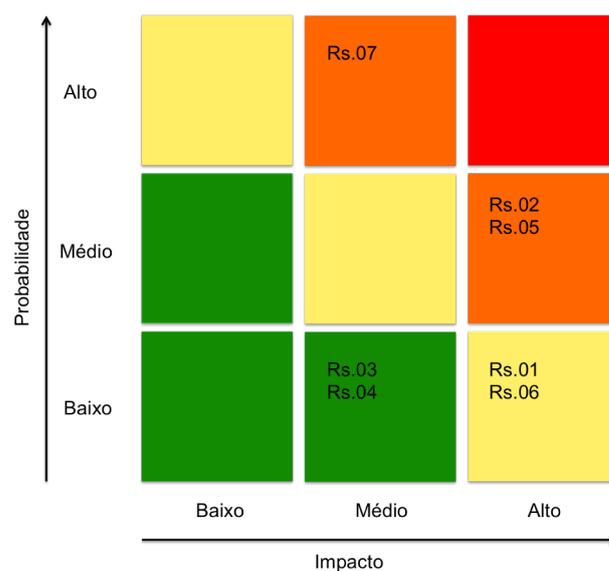


Figura 3.6: Matriz de exposição de riscos.

Rs.01 - Incapacidade para desenvolver algoritmo com desempenho adequado

Uma vez que algumas das funcionalidades introduzidas alteraram a abordagem anteriormente seguida pelo *Wizdee Discovery*, os resultados obtidos poderiam piorar.

Plano de Mitigação: Se os resultados referidos são da parte do algoritmo genético, então deveriam ser repensadas as *features* introduzidas e verificar uma a uma qual é que poderia estar a piorar os resultados do sistema. Por outro lado, se os resultados referidos fossem referentes ao contexto, deveria ser repensado o processo que estava a ser usado.

Ativação: Não foi necessário a ativação do plano de mitigação.

Rs.02 - Tempo despendido na execução de treino e testes

O tempo de computação necessário foi difícil de prever, o que poderia levar ao prolongamento do tempo inicialmente previsto.

Plano de Mitigação: A solução encontrada passava por dedicar uma máquina apenas às tarefas de treino e testes, levando a que o tempo de computação fosse menor.

Ativação: Não foi necessário a ativação do plano de mitigação.

Rs.03 - Restrições do software herdado

O software herdado poderia apresentar certas restrições à implementação do que foi inicialmente idealizado. Dado a *Wizdee* ter uma arquitetura complexa, poderia ser difícil a introdução de novas funcionalidades que não foram anteriormente refletidas.

Plano de Mitigação: Dependendo da restrição encontrada, deveria haver flexibilidade para a existência de pequenas alterações na arquitetura que ajudassem na introdução das *features* pretendidas.

Ativação: Foi necessário alterar diversos projetos de baixo-nível, de forma a conseguir propagar mais informação para as camadas superiores da arquitetura, onde ela era necessária.

Rs.04 - Dependência de terceiros

Na *Wizdee* o trabalho é realizado em equipa, pelo que certas tarefas podiam estar dependentes da execução de outra tarefa não desempenhada pelo estagiário, levando a desvios da meta traçada.

Plano de Mitigação: A monitorização contínua das tarefas devia ser feita, de modo a que ao detetar alguma modificação ao que foi inicialmente previsto pudesse ser logo contornada.

Ativação: Não foi necessário a ativação do plano de mitigação.

Rs.05 - Atraso no desenvolvimento de uma tarefa

Existiam tarefas que eram altamente dependentes de outras. O atraso numa dessas tarefas podia levar a que as dependentes se atrasem em efeito cascata.

Plano de Mitigação: Definição de um bom plano de escalonamento.

Ativação: Foi necessária a ativação deste plano, uma vez que a otimização do algoritmo genético demorou mais do que inicialmente previsto.

Rs.06 - Pouca experiência no desenvolvimento de algoritmos genéticos

O estagiário possuía pouca experiência no desenvolvimento de algoritmos genéticos, o que poderia causar atraso nas tarefas que envolviam o uso deste tipo de algoritmos.

Plano de Mitigação: Existem pessoas na empresa com experiência em algoritmos de aprendizagem máquina a quem o estagiário podia recorrer se encontrasse problemas.

Ativação: Não foi necessário a ativação do plano de mitigação.

Rs.07 - Impossibilidade de testar em ambiente real

Pretendia-se que os testes de personalização fossem realizados em ambiente real, uma vez que o contexto é algo difícil de validar já que varia de utilizador para utilizador. Desta forma, ao testar num ambiente real era possível obter resultados mais fiáveis. No entanto, não tendo uma solução suficientemente estável e testada é impossível a sua introdução no produto final fornecido a clientes.

Plano de Mitigação: Os testes deveriam ser realizados pela equipa de testes existente na empresa, a qual está ambientada ao produto, mas não tem conhecimentos de *backend* do sistema. Os diferentes elementos da equipa de testes deveriam possuir diferentes contextos.

Ativação: O plano de mitigação foi ativado, sendo que os testes foram realizados pela equipa de testes.

Capítulo 4

Implementação

Este capítulo pretende dar a conhecer ao leitor os processos que foram adotados e os métodos que foram implementados, de modo a serem cumpridos os requisitos descritos anteriormente.

4.1 Colaboração

A implementação da colaboração pode ser dividida em quatro grandes áreas: partilha interna, partilha externa, suporte para comentários e suporte para notificações. As estruturas criadas, bem como as funcionalidades, serão alvo de atenção nas secções seguintes. É de salientar que no âmbito deste trabalho não serão desenvolvidas as interfaces de utilizador necessárias à utilização das funcionalidades implementadas. A *Wizdee* dispõe de uma equipa dedicada exclusivamente ao desenvolvimento do interface de utilizador do produto. Uma tabela com os *endpoints* criados na API Rest pode ser visualizada nos anexos - tabela A.1, dando uma noção da implementação realizada para cada uma das áreas da colaboração.

4.1.1 Partilha Interna

A partilha interna foi implementada sobre os métodos já existentes de gestão de relatórios e *dashboards*, como a adição, edição, remoção e obtenção.

Começou-se por fazer a alteração do esquema da base de dados, representado na figura 4.1, onde foi criada a tabela *workspace_share*, a qual armazena a informação de partilha, fazendo a correspondência entre o objeto e o utilizador/grupo com quem foi realizada a partilha.

Sempre que um objeto é criado ou editado, verifica-se se houve alguma alteração às partilhas realizadas, cuja informação chega na forma de duas estruturas - *SharedUsers* e *SharedGroups*. Em caso afirmativo, são adicionados os registos correspondentes à partilha na tabela referida. Para isto ser possível, os comandos de adição e edição existentes foram alterados.

Os comandos para a obtenção dos objetos requeridos foram os que sofreram maiores alterações. Estes tiveram que realizar a união de diferentes tabelas, de modo a obter tanto os objetos do utilizador, como os partilhados com ele. Para serem obtidos apenas os objetos partilhados com um utilizador ou apenas aqueles que foram criados por si, o pedido é acompanhado de um parâmetro booleano capaz de fornecer essa informação.

Quando existe a remoção de algum livro, página, *widget* ou *dashboard*, os registos da tabela *workspace_share* são também removidos.

Importante referir, que a segurança foi algo que foi tido em conta, uma vez que foram

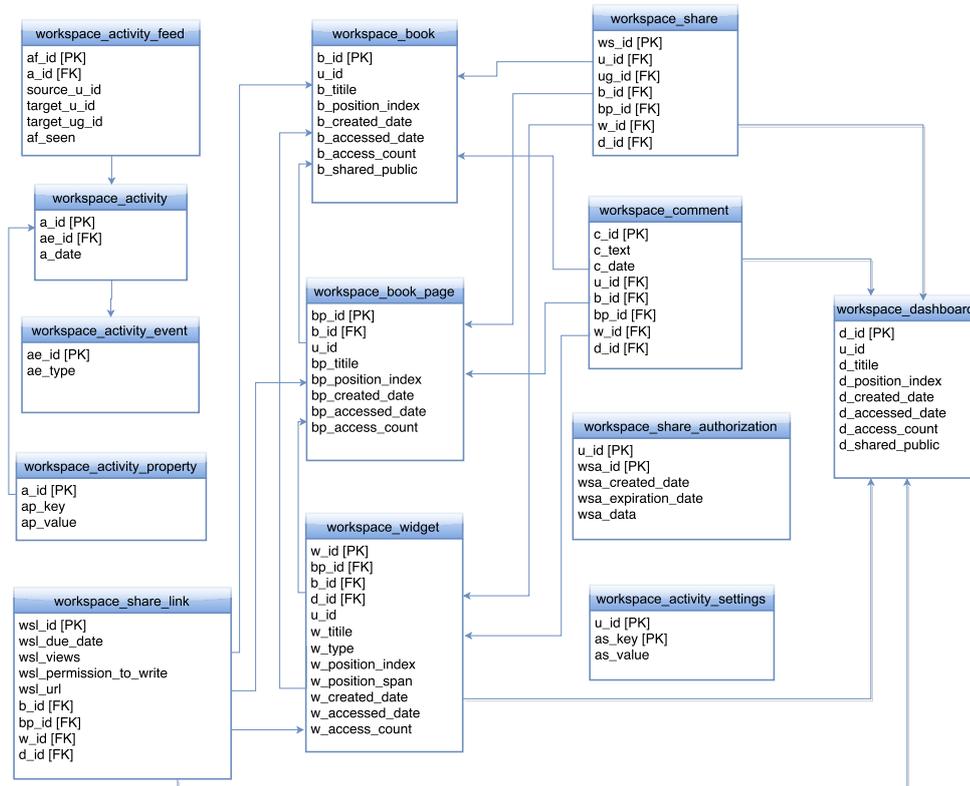


Figura 4.1: Diagrama da base de dados.

implementadas medidas que apenas os utilizadores proprietários dos livros, páginas ou *widgets* e aqueles com quem foram partilhados têm acesso ao conteúdo partilhado.

As funcionalidades de partilha interna já se encontram integradas no produto e em fase de produção, pelo que já são usadas por clientes. Na figura 4.2 é possível observar o processo para a partilha de *widgets* entre grupos e na figura 4.3 o processo para a partilha de *widgets* entre utilizadores.

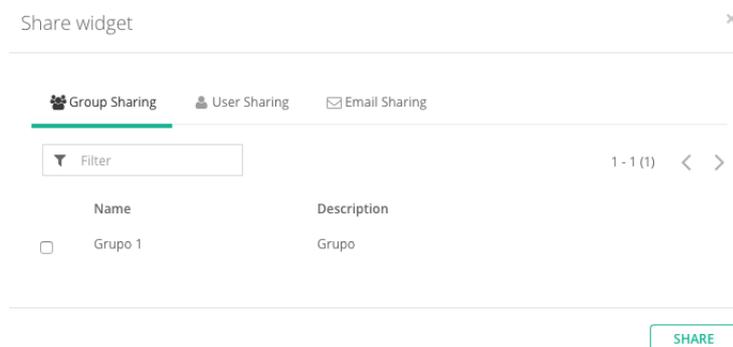


Figura 4.2: Partilha de *widgets* entre grupos.

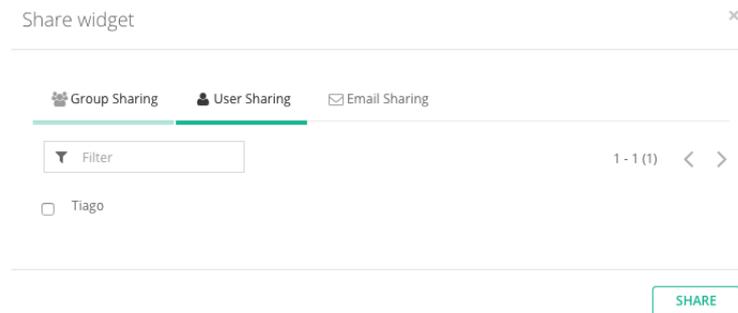


Figura 4.3: Partilha de *widgets* entre utilizadores.

4.1.2 Partilha Externa

A partilha externa foi dividida em quatro tipos de partilhas: partilha por *link*, partilha por email, partilha para redes sociais e partilha para ferramentas de produtividade. Estas irão ser descritas nas secções seguintes.

Partilha por *Link*

Na partilha por link é possível a partilha de livros, páginas, *widgets* ou *dashboards*. Neste tipo de partilha pode ser definido o tempo de validade do *link*, sendo que uma vez expirado, deixa de estar disponível. Outra das propriedades que este tipo de partilha pode ter é ao nível das permissões. Estas podem ser de escrita ou apenas de leitura, podendo no primeiro caso o utilizador que recebe o *link* editar os dados que foram partilhados. No segundo caso, este utilizador pode apenas visualizar os dados.

Através do pedido de geração do *link*, é criado um registo na tabela *workspace_share_link* com um URL único, gerado através das propriedades recebidas, usando para isso o algoritmo de *hash* MD5¹.

O utilizador pode, depois de partilhar um *link*, revogar o acesso ao mesmo. Quando esse pedido é efetuado, o registo respeitante a este *link* é removido da base de dados e deixa de estar acessível.

Quando os utilizadores com quem foi partilhada a informação tentam aceder ao sistema através desse *link*, é feito um pedido ao sistema e este devolve os objetos a que o utilizador tem acesso.

Para a verificação da expiração de um *link*, foi implementado um mecanismo usando *Quartz-Scheduler*², o qual permite agendar uma tarefa e esta ser executada nesse momento. Assim, foi definido que de hora a hora, esta verificação fosse feita, e caso a data atual seja superior à data de expiração, o link é removido e deixa de estar disponível.

Partilha por Email

A partilha por email apenas se encontra disponível para a partilha de *widgets*. Para ser possível o envio de um email foi usada a *JavaMail API* disponibilizada pela *Oracle*³, a qual fornece uma estrutura independente da plataforma e do protocolo, possibilitando a construção de um email e o seu envio⁴.

Quando um utilizador realiza o pedido, apenas tem que definir os destinatários, o assunto e o texto que quer enviar. Como anexo, é enviado o *widget* que o utilizador quer

¹MD5 - Algoritmo de *hash* de 128 bits unidirecional criado pela RSA Data Security, Inc..

²*Quartz Scheduler* - <http://www.quartz-scheduler.org>

³*Oracle* - <http://www.oracle.com>

⁴*JavaMail API* - <http://www.oracle.com/technetwork/java/javamail/index.html>

partilhar, podendo este ser visualizado em formato PDF, PNG ou SVG. Recebida esta informação, o sistema constrói o email usando a JavaMail API e este é enviado para os destinatários pretendidos.

As partilha por email já se encontra integrada no produto e em fase de produção, pelo que já é usada por clientes. Na figura 4.4 pode-se observar o processo usado para a partilha de um *widget* por email.

The screenshot shows a web interface titled "Share widget" with a close button (x) in the top right corner. Below the title, there are three tabs: "Group Sharing", "User Sharing", and "Email Sharing". The "Email Sharing" tab is currently selected and highlighted with a green underline. Below the tabs, there are three input fields: "To" (a single-line text box), "Subject" (a single-line text box), and "Text" (a larger multi-line text area). At the bottom right of the form, there is a green button labeled "SHARE".

Figura 4.4: Partilha de *widget* por email.

Partilha para Redes Sociais

As redes sociais escolhidas para a implementação da partilha para redes sociais foram o *Facebook*, o *Twitter* e o *LinkedIn*.

Começou-se pelo processo de autenticação, seguindo para isso o modelo *OAuth*, cujo processo pode ser visualizado na figura 4.5. Este é iniciado por um pedido de um *request token* por parte do consumidor ao fornecedor de serviço, o qual é imediatamente devolvido, direcionando o utilizador para uma página de autenticação do serviço em causa. Realizada a autenticação, é pedido ao utilizador que confirme que a aplicação pode ter as permissões indicadas. Em caso de confirmação, o utilizador é redirecionado novamente para o sistema de origem, e é realizado um pedido do *access token*. Uma vez obtido o *access token*, este é guardado na estrutura *workspace_authorization*, tendo o cuidado desta informação ser devidamente encriptada. Posteriormente, é possível o acesso a recursos protegidos, permitindo à plataforma partilhar *widgets* na rede social à escolha do utilizador.

Na tabela 1, a qual se encontra em anexo, foi apenas referenciado um *endpoint* nos diferentes pedidos de *tokens*. Isto acontece porque os outros pedidos terão que ser realizados ao nível da interface com o utilizador, comunicando com *servlets* responsáveis por estas ações.

Este tipo de partilha apenas permite a partilha de *widgets*. Desta maneira, o utilizador consegue partilhar esta informação através de uma imagem, podendo adicionar uma descrição.

Na implementação da partilha para o *Facebook* foi usada uma biblioteca Java *open-source - Facebook4J*⁵, a qual permite o uso da API desta rede social e oferece uma rápida

⁵*Facebook4J* - <http://facebook4j.org/>

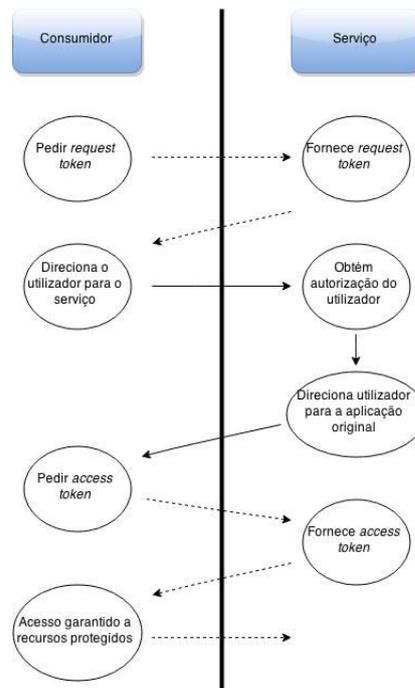


Figura 4.5: Diagrama de autenticação *OAuth*.

integração com a aplicação. Da mesma maneira, foi usada uma biblioteca para a integração do *Twitter* - *Twitter4J*⁶. Para o *LinkedIn* foi ainda utilizada uma outra biblioteca Java - *Scribe*⁷, a qual facilita a execução de pedidos *OAuth*.

Nos *endpoints* apresentados pode-se observar que o *Facebook* possui mais dois que as outras redes sociais. Isto, porque aqui é possível a criação de álbuns de fotografias, para onde o utilizador poderá enviar a sua partilha, podendo também escolher um álbum já existente.

Existe ainda um *endpoint* que permite verificar se o *access token* existente na base de dados ainda se encontra válido, uma vez que o utilizador pode proibir a aplicação de aceder ao seu perfil ou, no caso do *LinkedIn*, o próprio *access token* tem uma validade de 60 dias, pelo que o utilizador deve voltar a dar permissão à aplicação para aceder aos recursos protegidos.

Partilha para Ferramentas de Produtividade

A partilha para ferramentas de produtividade é semelhante à partilha para redes sociais, na medida em que o processo de autenticação é igual, usando também o modelo *OAuth* representado na figura 4.5, tendo sido igualmente implementados os *endpoints* respeitantes à geração do *access token* e à sua validação.

Na implementação da integração com a *Dropbox* foi usada uma biblioteca disponibilizada pela própria *Dropbox*⁸, permitindo a adição de documentos na conta do utilizador, numa pasta denominada de *Wizdee*. Podendo o utilizador partilhar *dashboards*, livros, páginas e *widgets*, sendo a organização destes na *Dropbox* feita de forma diferente. Assim, no caso de um *dashboard* ou livro ser partilhado, este é criado na raiz dessa pasta. No caso de ser uma página, é criada uma pasta com o nome do livro correspondente e o ficheiro

⁶ *Twitter4J* - <http://twitter4j.org/>

⁷ *Scribe* - <https://github.com/fernandezpablo85/scribe-java>

⁸ *Dropbox* - <https://www.dropbox.com/developers/core/sdks/java>

é criado dentro desta. No caso de ser um *widget* e este pertencer a um relatório é criado uma pasta com o nome do livro, seguida da criação de uma pasta com o nome da página correspondente, e a imagem representante do *widget* é criada nesta última. Por outro lado, se o *widget* pertencer a um *dashboard* é criada uma pasta com o nome do *dashboard* e a imagem do *widget* é criada dentro desta pasta.

Para a criação de notas, foi implementada a conexão com o *Evernote*, usando um SDK para Java desta ferramenta, o qual contém o código necessário para as chamadas à *Evernote Cloud API*. Através destas chamadas é possível a criação de blocos de notas, a obtenção dos blocos já existentes e a criação de notas. As notas aqui referidas são sempre referentes a um determinado *widget*, pelo que ao ser criada uma nota, esse *widget* é representado através de uma imagem e acompanhado por um texto à escolha do utilizador. Esta nota pode ser criada num bloco de notas escolhido pelo utilizador.

Com o *Google Calendar*, o utilizador tem a opção de criar um evento no seu próprio calendário. Para isto, foi usada uma biblioteca disponibilizado pela *Google* para a linguagem Java. Através desta, foi possível a criação dos eventos, podendo conter informação como o seu sumário, localização, descrição e emails dos vários intervenientes, para os quais serão enviados os convites. Como esta criação está diretamente relacionada com os dados visualizados no *Wizdee Discovery*, na descrição é adicionado um URL que contém a imagem do *widget*, de modo a contextualizar todos os intervenientes, ou mesmo o próprio utilizador. Reunida a informação necessária, o utilizador pode marcar o evento para uma data sobreposta com outro. Por este motivo, é dada uma notificação ao utilizador, avisando-o da sobreposição, dando-lhe a oportunidade de mudar a data do evento a ser criado. Caso seja da sua vontade que exista esta sobreposição, o evento é criado.

Por último, a integração com o *Trello* possibilitou a anotação de tarefas. Ao contrário das anteriores, este não disponibiliza bibliotecas para a sua integração, pelo que os pedidos HTTP foram implementados de raiz. A organização de tarefas do *Trello* pode ser visualizada na figura 4.6. Os quadros são a estrutura principal, os quais podem conter diferentes listas e estas vários cartões, sendo estes os representantes das tarefas. Os métodos criados permitiram a criação e obtenção de todos os elementos desta estrutura.

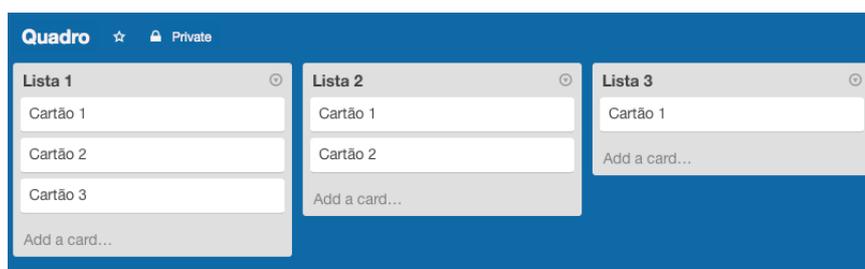


Figura 4.6: Organização de tarefas no Trello.

4.1.3 Suporte para Comentários

A implementação do suporte para comentários permite ao utilizador adicionar comentários a um livro, página, *widget* ou *dashboard*. Para isso ser possível, foram criados quatro *endpoints*, os quais permitem a adição, remoção, edição e obtenção destes.

Tendo a informação relativa ao comentário, um registo é adicionado na tabela *workspace_comment*, através de um comando criado para esse efeito. Este registo possui informação de onde foi realizado o comentário, através do identificador do objeto, quem o fez, o texto e a data da sua criação.

Para a remoção de um comentário, basta ser dado o identificador deste e o seu registo é removido da base de dados.

Um comentário pode ainda ser editado, pelo que se já existe um comentário com aquele identificador na base de dados, este é atualizado para a sua versão mais recente.

Por último, para obter estes comentários, também foi criado um comando que, dado o identificador do livro, página, *widget* ou *dashboard*, são obtidos os registos referentes a este, tendo como resposta uma estrutura com a informação referida acima.

Os comentários permitem ainda a adição de menções, as quais podem representar utilizadores ou referências a outros livros, páginas, *widgets* ou *dashboards*. Para isto, a interface tem a responsabilidade de gerar um XML capaz de identificar o tipo de menção realizada, bem como o respetivo identificador. O sistema, posteriormente, através da utilização do *XML DOM Parser*⁹, consegue extrair a informação da menção que foi realizada.

4.1.4 Suporte para Notificações

As notificações têm como função avisar o utilizador sempre que algo é partilhado com ele, ou avisar o utilizador sempre que é criado um comentário num relatório ou *dashboard* seu ou ele próprio é mencionado num comentário. Assim, podem existir cinco tipo de eventos:

- *Share Book* - Evento correspondente à partilha de um livro;
- *Share BookPage* - Evento correspondente à partilha de uma página;
- *Share Widget* - Evento correspondente à partilha de um *widget*;
- *Share Dashboard* - Evento correspondente à partilha de um *dashboard*;
- *Comment* - Evento que ocorre quando é feito um comentário num objeto seu;
- *Mention* - Evento que ocorre quando o utilizador é mencionado num comentário.

Assim, sempre que existe uma partilha ou comentário e o utilizador esteja relacionado com estes, é adicionada uma entrada na base de dados com essa informação, criando um novo registo na *workspace_activity_feed*. De maneira a fazer a interligação com as restantes tabelas como a *workspace_activity_event*, a qual representa o tipo de evento realizado, e a *workspace_activity_property*, dando a informação das propriedades a que a notificação diz respeito, existe uma tabela de ligação, a *workspace_activity*, a qual contém ainda a data da sua criação.

Para o acesso a estas notificações foi criado um *endpoint* (Ep.51), o qual recebendo o identificador do utilizador é devolvida uma listagem com todas as suas notificações.

O utilizador pode ainda editar as definições relativas às notificações que recebe. Para isso foi implementado um método que permite definir se o utilizador quer receber ou não, guardando essa informação na tabela *workspace_share_settings*.

O suporte para notificações já se encontra integrada no produto e em fase de produção, pelo que já é usado por clientes, sendo estas notificações exibidas como mostra a figura 4.7.



Figura 4.7: Notificações do utilizador.

⁹ *XML DOM Parser* - biblioteca que contém os métodos necessários percorrer um XML e poder extrair informação, bem como adicionar, editar e remover nós.

4.2 Ordenação de Cenários

A importância da ordenação de cenários já foi referida anteriormente, uma vez que a implementação da personalização assenta sobre manipulação da pontuação atribuída a cada cenário. Para isto foi usada a abordagem que já era seguida no *Wizdee Discovery*, a qual se baseia no uso de um algoritmo genético. Assim, numa primeira fase, foi necessário existir uma adaptação e otimização deste e, numa segunda fase, aplicar a informação extraída do AG no motor de pesquisa para uma melhoria na ordenação dos cenários.

De seguida, irão ser referidas várias vezes as *features* que foram introduzidas no algoritmo genético, bem como no motor de pesquisa do *Wizdee Discovery*. Estas, devido a questões de confidencialidade da empresa, foram remetidas para o anexo C, onde está presente a descrição de cada uma, sendo estas divididas em dois tipos: genéricas e de domínio. As primeiras são aquelas que são independentes do domínio, e, por outro lado, as segundas são calculadas com base no domínio e na relevância de cada uma para o utilizador. Estas últimas são importantes na implementação da personalização, como será descrito na secção 4.3.

4.2.1 Adaptação do Algoritmo Genético

O algoritmo genético usado na abordagem já seguida pela empresa tinha as seguintes componentes:

- Dados de Entrada - Foram gerados um conjunto de aproximadamente 500 pesquisas, tendo sido selecionado o cenário mais correto para cada uma delas e de seguida gravadas numa base de dados.
- Representação - Cada indivíduo na população é constituído por um *array*, onde cada posição representa os pesos correspondentes às *features* implementadas, sendo que um destes *arrays* representa a solução final.
- Função de Avaliação - Calcular quantos cenários existem que são melhores que o atual cenário. Quanto mais próximo de zero melhor.
- Seleção - Método roleta.
- Operadores Genéticos:
 - Cruzamento - Cruzamento por um ponto. É escolhida aleatoriamente uma posição do array de dois indivíduos e dois descendentes são gerados. O primeiro possui o array do primeiro progenitor desde o início até ao ponto escolhido, sendo o resto do array completado com do segundo progenitor desde o ponto até ao fim. O segundo descendente é o inverso. Uma ilustração deste processo é representado na figura 4.8.

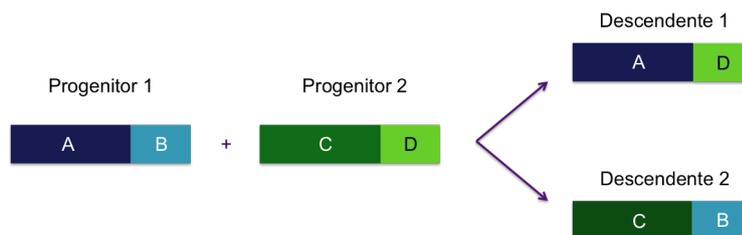


Figura 4.8: Cruzamento por um ponto.

- Mutação - É escolhido aleatoriamente um x número de mutações a serem realizadas. Após esta escolha, em cada indivíduo são escolhidos x valores aleatórios e é subtraído ou adicionado um valor aleatório no intervalo $[-1,1]$.
- Inserção - Geracional. Quando o tamanho da população de descendentes é igual ao da população de progenitores, esta é substituída pela dos descendentes.
- Condição de Paragem - Valor de *fitness* atingir zero, significando que todos os melhores cenários estavam na primeira posição.

Este algoritmo foi adaptado para ser usado juntamente com o novo motor de pesquisa do *Wizdee Discovery*, pois entretanto este foi alterado para poder responder a necessidades que o antigo era incapaz. Para isso, foi alterada a forma como o conjunto de dados de entrada estava a ser usado, porque estes tinham sido gerados recorrendo ao motor de pesquisa antigo. Assim, foi necessário que estes se tornassem compatíveis com o novo motor para poderem ser reavaliados e atualizados de acordo com as novas funcionalidades deste. Para além destas alterações e devido à crescente complexidade dos modelos de entrada e aumento da ambiguidade, a condição de paragem teve que ser modificada de forma a que o algoritmo corra em 20.000 gerações, em vez de terminar quando o valor de *fitness* atinge zero, como acontecia na abordagem anterior.

4.2.2 Otimização do Algoritmo Genético

O processo anterior usava *features* que já se encontravam desatualizadas, pois já não garantiam a obtenção de bons resultados (capítulo 5) quando aplicados em ambientes reais, neste caso no âmbito do conetor para um CRM popular, o serviço *Salesforce*¹⁰. Estas tiveram que ser substituídas por novas, as quais foram debatidas e pensadas com base no tipo de pesquisas realizadas pelos utilizadores sendo depois adicionadas ao conjunto de dados de entrada. A maior dificuldade na introdução destas novas *features* foi a falta de informação necessária para implementar as mesmas. Para contrariar isto foi necessário alterar diversos projetos de baixo-nível, de forma a conseguir propagar mais informação para as camadas superiores da arquitetura, onde ela era necessária.

Além disso, para que estas novas *features* fossem implementadas e facilmente integráveis no novo motor o processo de extração destas teve que ser alterado, para ficar de acordo com o comparador de cenários usado no *Wizdee Discovery*.

O processo para implementação destas *features* está representado na figura 4.9. Num momento inicial existia a ideia de uma nova *feature*, sendo que de seguida era necessário implementar o processo de extração. Se esta extração fosse direta então a informação podia passar à fase seguinte, se não eram implementadas novas funcionalidades noutros projetos de baixo-nível de modo a ter acesso à informação necessária no momento da extração, passando a ser uma extração direta. A fase seguinte prende-se com normalização do valor da nova *feature* para o intervalo $[0,1]$, de modo a que não existam *features* com demasiado peso em relação às outras. Tendo o valor normalizado, este era introduzido na equação final de avaliação do cenário. De modo a ver a influência desta nova *feature*, um novo teste era executado. Caso o resultado fosse positivo, a *feature* era aproveitada. Caso contrário, esta era descartada. Este processo foi repetido inúmeras vezes para chegar à lista atual de *features*. Devido a este facto, o tempo despendido para esta fase foi maior do que o inicialmente previsto. Para as ideias das *features* eram vistos quais as pesquisas que estavam a falhar, analisando as diferenças existentes entre o melhor cenário e os que

¹⁰*Salesforce* - Produto de gestão da relação com o consumidor (CRM) na *cloud* - <http://www.salesforce.com>

estavam acima deste. Era necessário encontrar uma característica capaz de distinguir o melhor cenário dos restantes e fizesse aumentar o peso deste para se aproximar da primeira posição na listagem de cenários.

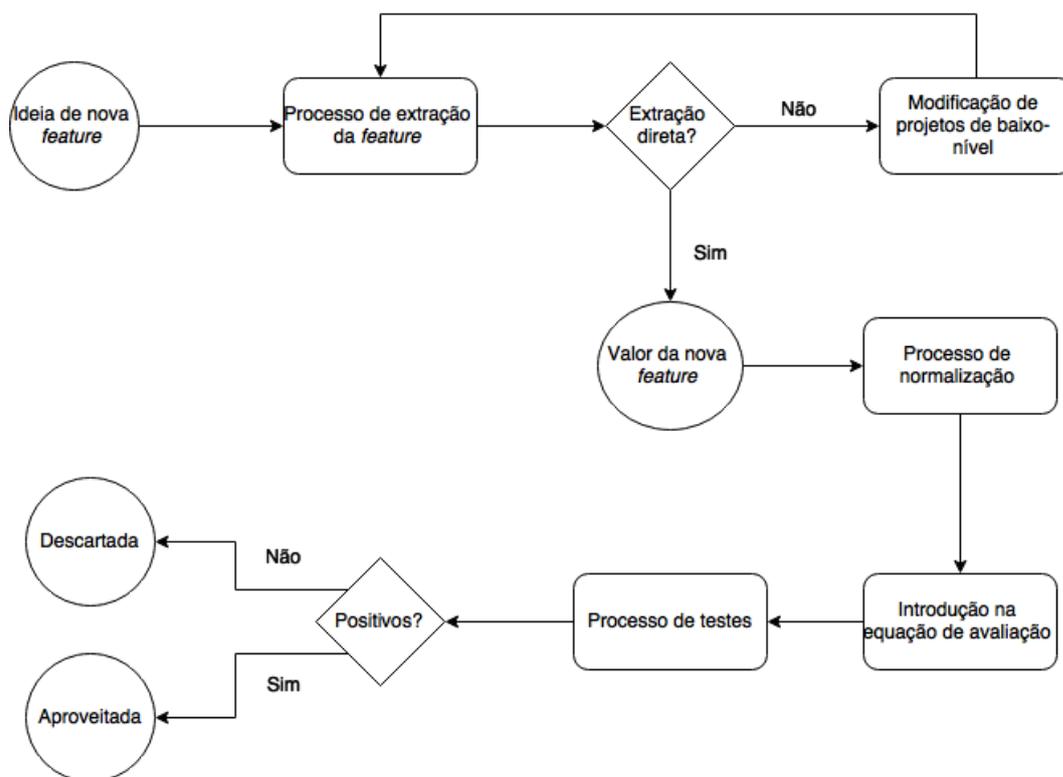


Figura 4.9: Processo para a implementação das *features*.

No final e decorridas as 20.000 gerações, o resultado final é um conjunto de pesos, onde cada um corresponde à importância que se deve dar a cada *feature* introduzida, as quais se encontram no anexo C. A soma destes pesos é sempre igual a 1.

4.2.3 Aplicação

Os resultados obtidos pelo algoritmo genético foram aplicados no método de ordenação de cenários existente no motor de pesquisa do *Wizdee Discovery*. Este processo calcula os valores das *features*, sendo que no final cada um destes é multiplicado pelo peso respetivo calculado pelo AG. Assim, é obtida a equação 4.1, a qual apresenta uma equação linear, onde a soma dos valores da multiplicação dos pesos (P) pela respetiva *feature* (F) apresenta a pontuação (S) de um cenário.

$$S = \sum_{i=1}^n (P_i \times F_i) \quad (4.1)$$

Posteriormente, os cenários são ordenados pela sua pontuação de forma descendente. Esta ordenação, embora melhorando de forma significativa a seleção do melhor cenário a ser apresentado ao utilizador, não garante ainda assim que o melhor cenário esteja em primeiro lugar, devido à grande ambiguidade que existe neste tipo de sistemas. Assim, uma tabela ou coluna pode ser mais importante para um utilizador, mas para outro poderá não o ser. É este o problema que a personalização tenta ultrapassar.

Esta nova versão do método de avaliação de cada cenário e graças aos testes alcançados (capítulo 5) já se encontra integrada no produto e em fase de produção.

4.3 Personalização

A personalização do sistema baseia-se na introdução de contexto, o qual varia consoante as interações do utilizador com o *Wizdee Discovery*. Desta forma, foram definidos dois tipos de contexto: o contexto de longo prazo e o contexto de curto prazo. O primeiro processa-se de forma *offline* e baseia-se na informação que o utilizador guarda nos seus *dashboards*. O segundo é realizado em tempo real, consoante as interações do utilizador com o sistema e as pesquisas realizadas. Assim, os resultados exibidos podem ser diferentes ao longo do uso da aplicação, mesmo que as pesquisas sejam repetidas, devido à alteração do contexto do utilizador.

Já foi referido anteriormente que as novas *features* podem ser divididas em genéricas e de domínio. Estas últimas baseiam-se no peso que cada tabela, coluna e relação entre tabelas têm para o utilizador. Usando a mesma abordagem do AG, foi possível o cálculo do peso de cada constituinte do domínio. Para isso, estas estruturas foram usadas como *features* e o algoritmo correu igualmente durante 20.000 gerações para o cálculo do peso de cada uma, sendo que estes pesos variam no intervalo de $[0,100]$. Calculados desta forma, os pesos do domínio refletem de forma genérica a importância de cada elemento do domínio. Mas, neste caso, pretende-se que estes pesos sejam depois personalizados para cada utilizador através da análise do seu contexto.

O contexto varia consoante os *dashboards* guardados pelo utilizador e as suas interações com o sistema. Assim, é atribuído um valor de variação de contexto para cada uma das fontes de informação usadas, dependendo do grau de confiança da informação que se pode extrair. Desta forma, quanto maior essa variação, maior o grau de confiança atribuído e maior influência do contexto nos pesos das tabelas, colunas e relações.

4.3.1 Contexto de Longo Prazo

O contexto de longo prazo tem um processo *offline*, o qual decorre diariamente à hora que for previamente determinada, sendo este realizado preferencialmente em horário noturno, de forma a não prejudicar a performance do sistema. Este evento é disparado recorrendo ao *Quartz-Scheduler*¹¹, percorrendo todos os utilizadores do sistema, usa os seus *dashboards* para influenciar o contexto de longo prazo. Consequentemente, consegue-se uma fonte fiável de informação, uma vez que os *dashboards* criados pelo utilizador representam objetivamente um conjunto de informação relevante para o mesmo e que ele pretende consultar de forma regular, sendo atribuída uma variação de contexto de 5, como descrito na tabela 4.1.

ID	Nome	Descrição	Varição
FI.01	<i>Dashboards</i>	<i>Dashboards</i> criados pelo utilizador.	5

Tabela 4.1: Fontes de informação consideradas no contexto de longo prazo.

Processo

O processo para a influência do contexto de longo prazo é realizado através dos *widgets* de um determinado utilizador que estão presentes nos seus *dashboards*. Para cada *widget* usa-se o cenário que está selecionado e são extraídas as estruturas que fazem parte dele. A estas estruturas, ou seja, tabelas, colunas e relações, é atribuído um valor de variação de contexto positiva de cinco, uma vez que a informação usada transmite um grande

¹¹*Quartz Scheduler* - <http://www.quartz-scheduler.org>

grau de confiança. De seguida, adicionam-se estas estruturas e o valor da variação a um mapeamento temporário.

O cenário que está selecionado pode não ser o primeiro da lista de cenários caso o utilizador tenha precedido à alteração de cenário. Nestes casos faz-se uma comparação entre o cenário escolhido e os que se encontravam acima deste. Nesta comparação detetam-se as estruturas que marcam a diferença entre os cenários. Detetadas as diferenças, estas são igualmente adicionadas ao mapeamento temporário, sendo atribuído um valor de variação positivo às estruturas presentes no cenário selecionado e um valor negativo às estruturas dos cenários que se encontram acima. Por exemplo, um cenário traduz "*opportunities by created date*" e outro "*opportunities by closed date*". Nestes dois cenários as tabelas envolvidas são as mesmas (*Opportunity*), as relações entre tabelas são as mesmas, uma vez que existe apenas uma, mas as colunas usadas são diferentes. Um usa a coluna *created date* e outro a *closed date*. Neste caso, são adicionadas estas duas colunas ao mapeamento, sendo atribuído um valor de variação positivo ou negativo consoante a importância que se quer dar a essa coluna.

Obtido o mapeamento temporário das estruturas nas quais se deve aumentar ou diminuir a importância para o utilizador, este é sempre normalizado para o intervalo de $[0,1]$. A normalização é calculada usando uma função exponencial invertida, como a que se pode ver na equação 4.2, assegurando que a estrutura tenha um crescimento controlado, dependente de um coeficiente de aprendizagem (ca).

$$vn = \begin{cases} 1 - e^{-ca \times |x|} & \text{se } x \geq 0 \\ -(1 - e^{-ca \times |x|}) & \text{se } x < 0 \end{cases} \quad (4.2)$$

O coeficiente de aprendizagem apresentado é um valor pré-definido, o qual foi obtido com recurso a uma série de testes empíricos, que garante que a evolução do contexto tem uma evolução controlada, evitando que os pesos das estruturas se aproximem muito rapidamente do valor mínimo, 0, e do valor máximo, 100. Para este coeficiente foi definido o valor de 0.05, garantindo assim que o peso de uma estrutura não varie de zero para cem ou de cem para zero através de uma variação de contexto que não justifique essa alteração. Por exemplo, uma estrutura com peso 100 terá que ter uma variação de contexto de -100 para o seu novo peso ser de 0.

Calculado o valor normalizado de variação para cada estrutura presente no mapeamento, é necessário o cálculo do valor que se deve adicionar ou subtrair ao atual peso da tabela, coluna ou relação. Para isso, foi usada a equação 4.3, a qual tem em conta o peso atual que a estrutura possui (PE_i).

$$D(PE_i) = vn \times PE_i \quad (4.3)$$

No entanto, a equação 4.3 mostrou que permitia que o valor final do peso de uma estrutura fosse menor que zero ou maior que cem. Esta equação permitia ainda que os pesos maiores e os pesos menores evoluíssem de igual forma, fazendo com que os valores mais próximos de zero demorassem muito a subir. Foi necessária a introdução de uma nova componente na equação, que permite contrariar esta situação, sendo obtida a equação 4.4.

$$D(PE_i) = vn \times PE_i \times \left(1 - \frac{PE_i}{100}\right) \quad (4.4)$$

Ainda assim, a equação 4.4 fazia com que os pesos mais altos continuassem a descer muito lentamente. Assim foi necessária a distinção do uso entre duas equações. A equação 4.4 é usada quando o valor normalizado é positivo. Por outro lado, quando o valor é negativo é usada outra equação. Assim, surge-nos o sistema apresentado pela equação 4.5.

$$D(PE_i) = \begin{cases} vn \times PE_i \times \left(1 - \frac{PE_i}{100}\right) & \text{se } vn \geq 0 \\ vn \times PE_i \times \frac{PE_i}{100} & \text{se } vn < 0 \end{cases} \quad (4.5)$$

No final, o valor do novo peso (nw) é calculado usando a equação 4.6, na qual é somado o valor previamente calculado ao peso atual da tabela, coluna ou relação (PA_i).

$$nw_i = D(PE_i) + PA_i \quad (4.6)$$

Exemplo

Imagine-se um utilizador que possui um *dashboard*, o qual contém dois *widgets*:

1. *Opportunities by Date*. Cenários:
 - (a) *Opportunities by Closed Date*
 - (b) *Opportunities by Created Date*
2. *Accounts by Date*
 - (a) *Accounts by Created Date of Lead*
 - (b) *Accounts by Converted Date of Lead*
 - (c) *Accounts by Closed Date of Opportunity*
 - (d) *Accounts by Created Date of Opportunity*

O primeiro *widget* tem o cenário selecionado correspondente a *opportunities by created date*, sendo que na listagem de cenários este encontra-se na segunda posição (1-b). O segundo *widget* tem o quarto cenário (2-d) selecionado, *accounts by created date of opportunity*.

O processo do contexto de longo prazo é disparado e usa estes dois *widgets* para o cálculo do contexto de longo prazo, seguindo as seguintes etapas:

1. Mapeamento temporário das diferenças entre o cenário escolhido e os que estão acima na listagem. No primeiro *widget* a diferença está no tipo de data que é escolhido - *created date vs closed date*. No segundo, a diferença está no tipo de data que é escolhido e nas tabelas envolvidas. O mapeamento fica então com o aspeto mostrado na tabela 4.2.

Widget	Estrutura do Cenário Escolhido	Diferenças Mapeadas
1	Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>
2	Tabela <i>Opportunity</i>	Tabela <i>Lead</i>
2	Relação <i>Account-Opportunity</i>	Relação <i>Account-Lead</i>
2	Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i> ; Coluna <i>Created Date</i> da Tabela <i>Lead</i> ; Coluna <i>Converted Date</i> da Tabela <i>Lead</i>

Tabela 4.2: Mapeamento temporário das diferenças.

2. Aplicação dos valores de variação de contexto. Por cada mapeamento feito é aplicado um valor de variação positivo nas estruturas do cenário escolhido e nas diferenças detetadas é aplicado um valor negativo. Desta forma, o processo é realizado da seguinte forma:

- (a) Comparando o cenário 1 a) e 1 b). Aqui a diferença detetada é a *Created Date* da tabela *Opportunity* - *Closed Date* da tabela *Opportunity*. Assim, estas duas colunas devem ser adicionadas ao mapeamento temporário, onde a coluna do cenário 1 b) deve ser preferida em relação à outra, ficando o mapeamento temporário como demonstrado na tabela 4.3.

Estrutura	Varição
Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	+5
Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>	-5

Tabela 4.3: Mapeamento temporário depois da primeira comparação.

- (b) Comparando o cenário 2 a) e 2 d). Nesta comparação são detetadas três diferenças. A primeira na tabela usada (*Opportunity vs Lead*), a segunda na coluna (*Created Date* da tabela *Opportunity vs Created Date* da tabela *Lead*) e a terceira na relação (*Account-Opportunity vs Account-Lead*). Uma vez que a coluna *Created Date* da tabela *Opportunity* já se encontra no mapeamento, o novo valor de variação desta é somado ao anterior. O mapeamento fica então como descrito na tabela 4.4.

Estrutura	Varição
Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	+10
Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>	-5
Tabela <i>Opportunity</i>	+5
Tabela <i>Lead</i>	-5
Coluna <i>Created Date</i> da Tabela <i>Lead</i>	-5
Relação <i>Account-Opportunity</i>	+5
Relação <i>Account-Lead</i>	-5

Tabela 4.4: Mapeamento temporário depois da segunda comparação.

- (c) Comparando o cenário 2 b) e 2 d). À semelhança da comparação anterior são detetadas três diferenças. Estas diferenças apenas se diferenciam na comparação das colunas usadas (coluna *Created Date* da tabela *Opportunity vs Converted Date* da tabela *Lead*). Na tabela 4.5 é possível analisar as alterações feitas no mapeamento.

Estrutura	Varição
Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	+15
Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>	-5
Tabela <i>Opportunity</i>	+10
Tabela <i>Lead</i>	-10
Coluna <i>Created Date</i> da Tabela <i>Lead</i>	-5
Relação <i>Account-Opportunity</i>	+10
Relação <i>Account-Lead</i>	-10
Coluna <i>Converted Date</i> da Tabela <i>Lead</i>	-5

Tabela 4.5: Mapeamento temporário depois da terceira comparação.

- (d) Comparando o cenário 2 c) e 2 d). Na última comparação a única diferença detetada é na coluna usada (coluna *Created Date* da tabela *Opportunity* vs *Closed Date* da tabela *Opportunity*). O mapeamento final fica então com as estruturas e respetivas variações de contexto presentes na tabela 4.6.

Estrutura	Variação
Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	+20
Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>	-5
Tabela <i>Opportunity</i>	+10
Tabela <i>Lead</i>	-10
Coluna <i>Created Date</i> da Tabela <i>Lead</i>	-5
Relação <i>Account-Opportunity</i>	+10
Relação <i>Account-Lead</i>	-10
Coluna <i>Converted Date</i> da Tabela <i>Lead</i>	-5
Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>	-10

Tabela 4.6: Mapeamento temporário depois da quarta comparação.

3. Normalização dos valores do mapa usando a equação 4.2, com o número de *widgets* igual a 2. Na tabela 4.7 são apresentados os valores finais de variação de contexto e respetiva normalização.

Estrutura	Variação	Normalização
Tabela <i>Lead</i>	-10	$\approx -0,39$
Tabela <i>Opportunity</i>	+10	$\approx +0,39$
Relação <i>Account-Lead</i>	-10	$\approx +0,39$
Relação <i>Account-Opportunity</i>	+10	$\approx +0,39$
Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	+20	$\approx +0,63$
Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>	-10	$\approx +0,39$
Coluna <i>Created Date</i> da Tabela <i>Lead</i>	-5	$\approx -0,22$
Coluna <i>Converted Date</i> da Tabela <i>Lead</i>	-5	$\approx -0,22$

Tabela 4.7: Valores finais da variação de contexto e respetiva normalização.

4. Cálculo dos novos pesos. Para isso são usadas as equações 4.3 e 4.4 para calcular o valor que deve ser somado ao peso atual, sendo que o peso final é dado pela equação 4.6. Na tabela 4.8 estão calculados os novos pesos para cada estrutura.

Estrutura	Peso Atual	Valor a Somar	Peso Final
Tabela <i>Lead</i>	43,86	-7,57	36,29
Tabela <i>Opportunity</i>	35,67	+9,03	44,70
Relação <i>Account-Lead</i>	0,80	-0,003	0,797
Relação <i>Account-Opportunity</i>	0,20	+0,08	0,28
Coluna <i>Created Date</i> da Tabela <i>Opportunity</i>	44,06	+15,58	59,64
Coluna <i>Closed Date</i> da Tabela <i>Opportunity</i>	83,80	-27,63	56,17
Coluna <i>Created Date</i> da Tabela <i>Lead</i>	71,24	-11,23	60,01
Coluna <i>Converted Date</i> da Tabela <i>Lead</i>	54,28	-6,52	47,76

Tabela 4.8: Novos pesos para cada estrutura.

Neste exemplo é notória a importância que a tabela *Opportunity* e as estruturas relativas a esta começaram a ter para o utilizador. Desta forma, e com exceção para a coluna

Closed Date da tabela *Opportunity*, estas estruturas viram os seus pesos a serem aumentados, enquanto que as estruturas que se revelaram ter menos importância para o utilizador viram os seus pesos a serem diminuídos.

4.3.2 Contexto de Curto Prazo

O processo de influência do contexto de curto prazo é muito semelhante ao do contexto de longo prazo. A única diferença é que este é processado em tempo real consoante as interações do utilizador com o sistema e as pesquisas que vai realizando. Desta forma, foram definidos quatro eventos que permitem que o contexto seja influenciado quando algum destes é despoletado. A cada um destes eventos está associado um grau de confiança, pois alguns transmitem informação mais fiável daquilo que o utilizador deseja. Por exemplo, um utilizador ao realizar uma pesquisa pode não ter encontrado a informação que desejava. Mas se esse mesmo utilizador gravar o *widget* que é gerado, significa que era aquela informação que queria. Assim, o segundo caso tem uma variação de contexto maior do que a que existe no primeiro caso.

Os eventos considerados para a influência do contexto de curto prazo estão presentes na tabela 4.9. Para a consideração destes eventos existem alguns aspetos que devem ser tidos em conta. Assim quando um utilizador realiza uma pesquisa, apenas se pode usar o *widget* gerado quando o cenário selecionado é o último escolhido pelo utilizador. Para isto, apenas quando o utilizador realiza outra pesquisa é que podemos usar a anterior para influenciar o contexto, pois desta forma sabemos que o cenário escolhido foi o último que o utilizador visualizou, tendo este uma maior probabilidade de ser o correto. Nestes eventos nunca se pode assegurar que a influência do contexto esteja a ser corretamente aplicada, pois o utilizador pode nem chegar a ver o que desejava e essa informação ser usada para alterar o contexto.

ID	Evento	Variação
E.1	O utilizador realiza uma pesquisa	1
E.2	O utilizador muda o tipo de gráfico	2
E.3	O utilizador abre um <i>dashboard</i>	3
E.4	O utilizador grava um <i>widget</i>	5

Tabela 4.9: Eventos do contexto de curto prazo.

4.3.3 Aplicação

A introdução do contexto teve diferentes áreas de aplicação: ordenação de cenários, *autocomplete* e pesquisas sugeridas.

Ordenação de Cenários

Ao alterar o peso das tabelas, colunas e relações, o contexto do utilizador influencia a ordenação de cenários. As *features* de domínio permitem que a informação que o utilizador frequentemente procura tenha mais peso que outro tipo de informação e, consequentemente, os cenários que possuem essas estruturas subam em relação aos outros.

Autocomplete

O *autocomplete* do sistema é alimentado com duas fontes de informação: o *Schema-Manager* e o *Query-Manager*. A primeira fonte dá-nos informação do esquema de dados, ou seja, sobre as tabelas e colunas envolvidas. A segunda, o *Query-Manager* completa o

autocomplete com as pesquisas frequentes do utilizador e também com as pesquisas sugeridas.

As sugestões do *autocomplete* são exibidas segundo uma ordem pré-estabelecida. Assim, a ordem é a seguinte:

1. Frequentes, vindas do *Query-Manager*.
2. Sugeridas, vindas do *Query-Manager*.
3. Entidades, vindas do *Schema-Manager*.
4. Atributos, vindas do *Schema-Manager*.

Cada *manager*, ao dar as suas sugestões, pode ultrapassar o número máximo de sugestões permitidas no *autocomplete*, pelo que o sistema possui um algoritmo capaz de mostrar estas sugestões de forma equitativa. Este trabalho incidiu apenas na ordenação das sugestões que eram apresentadas pelo *Query-manager* e *Schema-manager*.

Desta forma, no *Query-Manager* foi criada uma estrutura adicional *context_query*, onde é guardado o cenário escolhido para cada uma das pesquisas frequentes e sugeridas. Sempre que o contexto sofre alteração, seja de longo prazo ou curto prazo, estes cenários são atualizados, bem como o valor de domínio associado. Estas sugestões são então ordenadas segundo este valor e enviadas para o *autocomplete*.

Da parte do *Schema-Manager* não foi necessária a introdução de informação numa estrutura adicional, pois esta informação está sempre presente no contexto. Assim, este *manager*, ao ter acesso ao peso de cada tabela, coluna e relação, ordena as suas sugestões de forma descendente.

Pesquisas Sugeridas

Da mesma forma que o *Query-Manager* ordena as sugestões de *autocomplete*, este para sugerir uma pesquisa usa o mesmo método. Usando a estrutura adicional que foi implementada, ordena as sugestões apresentadas segundo o seu valor de domínio de forma descendente.

Capítulo 5

Testes e Validação

Este capítulo contém os testes que foram realizados, bem como as suas respetivas análises. Assim, foram realizados testes sobre três áreas: colaboração e partilha (5.1), ordenação de cenários (5.2) e personalização (5.3).

5.1 Colaboração e Partilha

A área de colaboração e partilha, estando inserida numa vertente mais de engenharia, foi dividida em dois tipos de testes: testes unitários e testes de aceitação. Ao contrário dos primeiros, os testes de aceitação apenas puderam ser usados para validar algumas das funcionalidades implementadas, as quais estão integradas na interface. As restantes serão introduzidas mais tarde.

5.1.1 Testes Unitários

A realização dos testes unitários foi feita recorrendo a *user stories*. Estas são breves descrições de funcionalidades direcionadas na perspetiva do utilizador e são bastante utilizadas em metodologias de desenvolvimento ágil (Rasmusson, 2010). A preferência por estas deve-se também ao facto de permitirem o mapeamento direto para testes unitários. As *user stories* seguem uma estrutura própria, pelo que devem seguir um determinado modelo e responder a três questões essenciais:

- Enquanto <tipo_de_utilizador> (Quem a precisa?)
- Quero <ação> (O que é que ele quer?)
- Para <objetivo> (Porque é que ele o quer?)

No anexo B estão descritos os testes que foram realizados seguindo o modelo apresentado, os quais foram realizados em JUnit¹, a qual é uma *framework* usada para executar testes repetitivos.

5.1.2 Testes de Aceitação

Os testes de aceitação são testes não-automatizados que permitem verificar se as funcionalidades implementadas funcionam como o esperado para o utilizador, sendo diferentes dos anteriores, uma vez que são executados pelos próprios utilizadores no uso da aplicação.

¹JUnit - junit.org

Desta forma, algumas das funcionalidades ao serem integradas na interface do *Wizdee Discovery* puderam ser testadas numa primeira fase pela equipa de UX no momento da integração e, numa segunda fase, pela equipa de testes.

As funcionalidades que entraram para os testes de aceitação foram as seguintes:

- Partilha interna.
- Partilha por email.
- Suporte de notificações.

Os elementos que fizeram parte destes testes teceram diversos comentários e foram reportando alguns erros que foram aparecendo, o que permitiu o melhoramento destas funcionalidades. De seguida são mostrados alguns dos melhoramentos que foram feitos através da realização dos testes de aceitação:

- Correções na estrutura de email, de modo a exibir um HTML realizado pela equipa de UX.
- Correção nas notificações, de modo a não exibir notificações sobre dados já inexistentes.
- Notificações com identificadores corretos, mas com nomes trocados.

5.2 Ordenação de Cenários

Para a ordenação de cenários foram usados testes automáticos que já eram usados pela empresa para validar a solução anterior.

5.2.1 Testes Automáticos

Os testes automáticos foram feitos recorrendo ao conjunto de dados de entrada que foi usado no algoritmo genético, o qual contém 500 pesquisas. Para a validação das novas *features* foi feito um teste em que foram usadas todas as pesquisas do conjunto de dados de entrada. Este teste pode ser visualizado na figura 5.1, onde a área verde representa a percentagem de acerto no cenário correto e a área amarela diz que o cenário correto está na listagem mas não é o primeiro. O teste foi realizado seguindo três abordagens diferentes: a abordagem anterior, a abordagem nova sem domínio, ou seja, nenhuma tabela, coluna ou relação entre tabelas têm um peso associado, e na abordagem nova, onde estas estruturas já têm um peso associado. Estes pesos foram igualmente calculados pelo algoritmo genético.

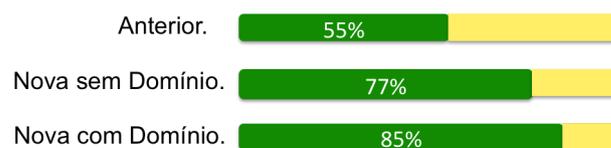


Figura 5.1: Testes automáticos com todas as 500 pesquisas.

5.2.2 Análise de Resultados

Analisando os testes da figura 5.1, verifica-se que na abordagem anterior em apenas 55% dos casos era escolhido o cenário correto. Com a introdução das novas *features* esta percentagem subiu para os 77% e, se se tiver em conta o domínio, consegue-se uma percentagem de acerto na ordem dos 85%.

São evidentes os melhoramentos que as novas *features* trouxeram ao *Wizdee Discovery*, podendo-se ainda concluir que o conhecimento do domínio tem uma influência muito positiva na identificação do cenário correto. Este conhecimento é importante na medida em que ao saber-se à partida quais as pesquisas que habitualmente são feitas, quais os dados que se querem ver e quais as relações de tabelas mais importantes, o sistema consegue reduzir muitos casos de ambiguidade que as suas interpretações podem ter. Este domínio está orientado aos utilizadores que geraram o conjunto de dados de entrada do algoritmo genético, pois o treino e testes foram realizados com o mesmo conjunto de dados.

Estes resultados mantiveram-se constantes ao longo dos vários testes realizados, pelo que são obtidos os mesmos resultados para o mesmo conjunto de pesquisas independentemente do número de testes executados.

Segundo o teste realizado para a validação das *features* que foram introduzidas para a ordenação de cenários pode-se verificar que vieram trazer uma melhoria significativa ao sistema, sendo que essa melhoria fixa-se nos 30% quando se compara a abordagem anterior com a abordagem nova com domínio.

5.3 Personalização

A personalização é algo que não é facilmente validada, uma vez que o contexto varia de utilizador para utilizador. Desta forma, foram pensados dois tipos de testes: quantitativos, onde foram extraídas métricas através da interação do utilizador com o sistema, e os testes qualitativos, onde foram recolhidas as opiniões de alguns utilizadores.

5.3.1 Testes Quantitativos

Os testes quantitativos foram recolhidos através da interação do utilizador com o sistema. Assim, foram recolhidas métricas da listagem de cenários e da listagem do *autocomplete*. Na primeira era vista a posição do cenário escolhido pelo utilizador e, na segunda, a posição no *autocomplete* da pesquisa que o utilizador realizou de seguida. Nesta último é importante realçar que o número máximo de sugestões apresentadas pelo *autocomplete* é de sete. Quando nos gráficos relativos a esta funcionalidade aparece o valor de oito significa que a pesquisa não se encontra na listagem de sugestões apresentadas.

Nestes testes foram também registadas métricas relacionadas com as pesquisas sugeridas, nas quais era retirada a posição da pesquisa que o utilizador escolhia, comparando sempre a abordagem com contexto com a abordagem sem contexto. No entanto, o estagiário preferiu não colocar essas métricas nestes testes, uma vez que a amostra era muito reduzida. Isto deve-se ao facto de os utilizadores que realizaram os testes estarem ambientados ao produto e aos dados, sendo que raramente se enganavam na pesquisa introduzida. Só nestes casos é que a pesquisa sugerida é apresentada. Mesmo assim, foi possível a validação destas indiretamente, porque também são apresentadas pesquisas sugeridas no *autocomplete*. Desta forma, a validação destas é realizada em conjunto com a validação da ordenação do *autocomplete*.

Para isto, foram realizados dois tipos de exercícios: um de contexto simulado e outro de exercícios com utilizadores. O primeiro foi realizado pelo estagiário, onde através de um conjunto de 30 pesquisas encadeadas mostra a influência do contexto na ordenação

dos cenários e *autocomplete*. O segundo foi realizado com a ajuda de três elementos da empresa, os quais não estão a par dos métodos de *backend*, mas encontram-se ambientados com o produto e o uso deste.

Contexto Simulado

Os testes de contexto simulado foram feitos de maneira a mostrar a influência do contexto. Para isto foram escolhidas trinta pesquisas encadeadas que proporcionassem a mudança do contexto e um impacto positivo deste na ordenação de cenários e *autocomplete*. As métricas usadas para estatística foram a posição do cenário escolhido pelo utilizador na listagem de cenários e a posição da pesquisa realizada pelo utilizador nas sugestões de *autocomplete*.

Na figura 5.2 pode ser visto a influência do contexto na ordenação de cenários. Quanto mais alto o valor da posição pior, uma vez que o cenário correto aparece cada vez mais distante da primeira posição. Desta maneira, é possível a comparação de onde o cenário correto estaria usando a abordagem anterior (série cinzenta) e usando a nova abordagem (série preta).

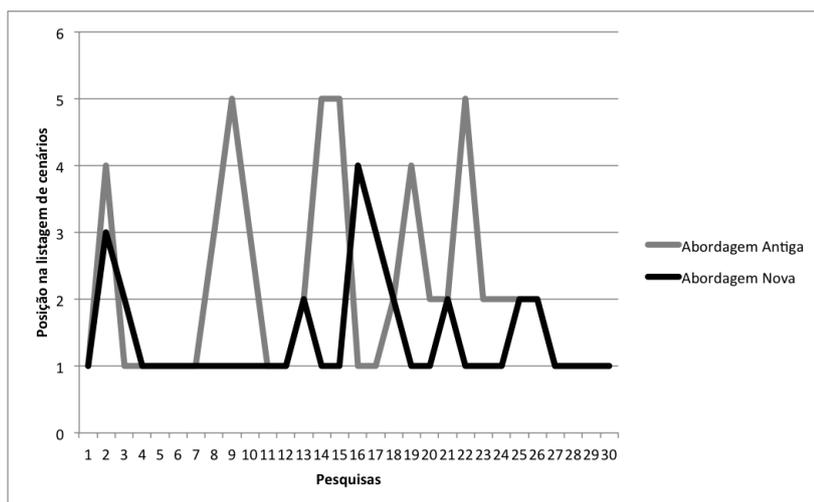


Figura 5.2: Testes quantitativos da ordenação de cenários por contexto simulado.

A outra parte dos testes incidiu no *autocomplete*, cujo o gráfico da figura 5.3 reflete a posição no *autocomplete* da pesquisa que o utilizador realizou de seguida. O utilizador pode nem ter clicado na sugestão apresentada no *autocomplete*, mas mesmo assim foi retirada a posição de onde a pesquisa que ele realizou se encontraria se tivesse usado a sugestão. Mais uma vez, a série cinzenta representa a posição pela abordagem anterior e a série preta a posição pela nova abordagem.

Neste exercício ainda foram retiradas métricas relacionadas com o tempo de geração de cenários, comparando o tempo usando a abordagem antiga e a abordagem nova. Esta comparação pode ser visualizada na figura 5.4.

Já na tabela 5.1 podem ser analisados alguns valores estatísticos relacionados com as métricas apresentadas anteriormente, onde é demonstrada a média, o desvio padrão e o erro associado ao intervalo de confiança de 95% de cada uma das abordagens para a ordenação de cenários, *autocomplete* e geração de cenários.

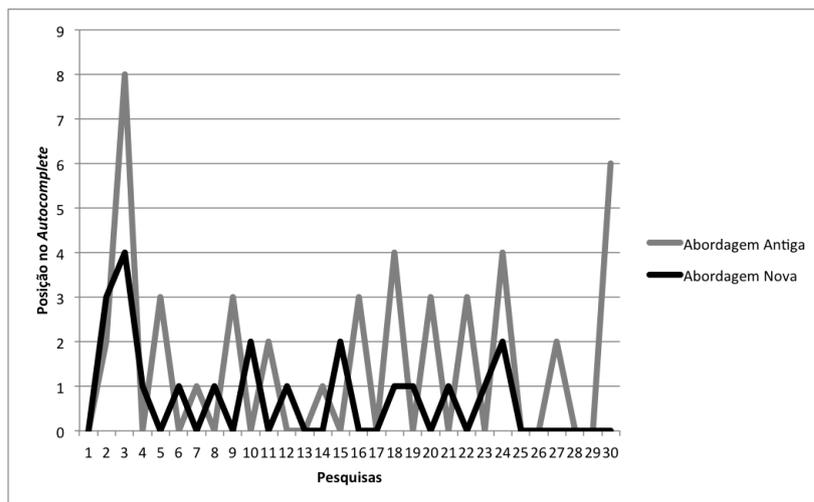


Figura 5.3: Testes quantitativos do *autocomplete* por contexto simulado.

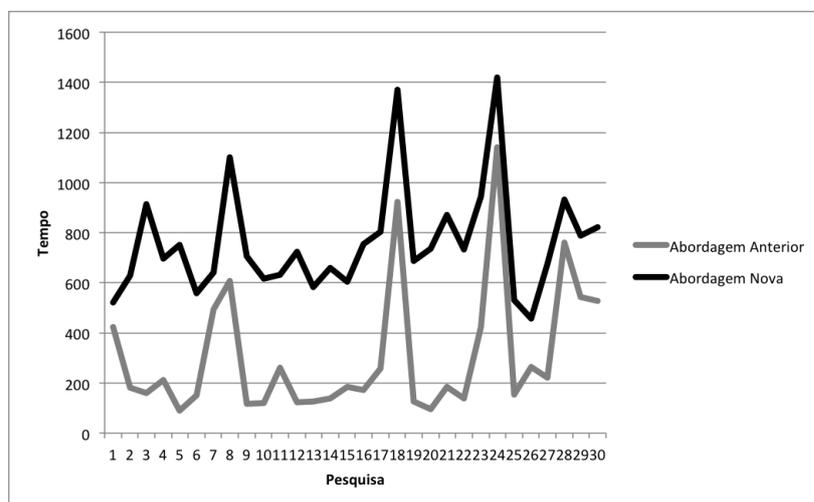


Figura 5.4: Comparação dos tempos de geração de cenário entre a abordagem antiga e a abordagem nova.

Exercício com Utilizadores

O exercício com utilizadores foi realizado com ajuda de três elementos da empresa, os quais não pertencem ao desenvolvimento do *backend*, mas sabem manusear o *Wizdee Discovery*. Desta maneira, pretende-se que este exercício se assemelhe muito a um exercício realizado por um utilizador normal do produto em ambiente real.

À semelhança do que ocorreu no exercício de contexto simulado, as métricas retiradas neste exercício foram as mesmas. Porém foi conseguido uma maior amostra, pois foram realizadas cem pesquisas. Na figura 5.5 pode ser visualizada a influência do contexto na ordenação de cenários e, mais uma vez, quanto maior a posição do cenário pior é o resultado. Desta maneira, a série cinzenta representa a abordagem anterior e a série preta a abordagem nova com a introdução do contexto.

Também muito parecido ao exercício de contexto simulado surge o gráfico da figura 5.6. Este representa a comparação entre a posição da pesquisa que o utilizador realizou nas sugestões de *autocomplete* entre a abordagem anterior e a nova.

Na tabela 5.2 podem ser vistos valores estatísticos relacionados com as métricas apre-

Teste	\bar{x}	σ	Erro com Intervalo de Confiança de 95%
Ordenação de Cenários na Abordagem Antiga	2,11	$\pm 1,39$	$\pm 0,02$
Ordenação de Cenários na Abordagem Nova	1,42	$\pm 0,74$	$\pm 0,01$
<i>Autocomplete</i> na Abordagem Antiga	1,54	$\pm 2,12$	$\pm 0,02$
<i>Autocomplete</i> na Abordagem Nova	0,64	$\pm 0,95$	$\pm 0,01$
Tempo na Geração de Cenários na Abordagem Antiga	311,43 ms	$\pm 270,74$	$\pm 3,10$
Tempo na Geração de Cenários na Abordagem Nova	775,82 ms	$\pm 223,24$	$\pm 2,56$

Tabela 5.1: Estatísticas associadas aos testes com contexto simulado.

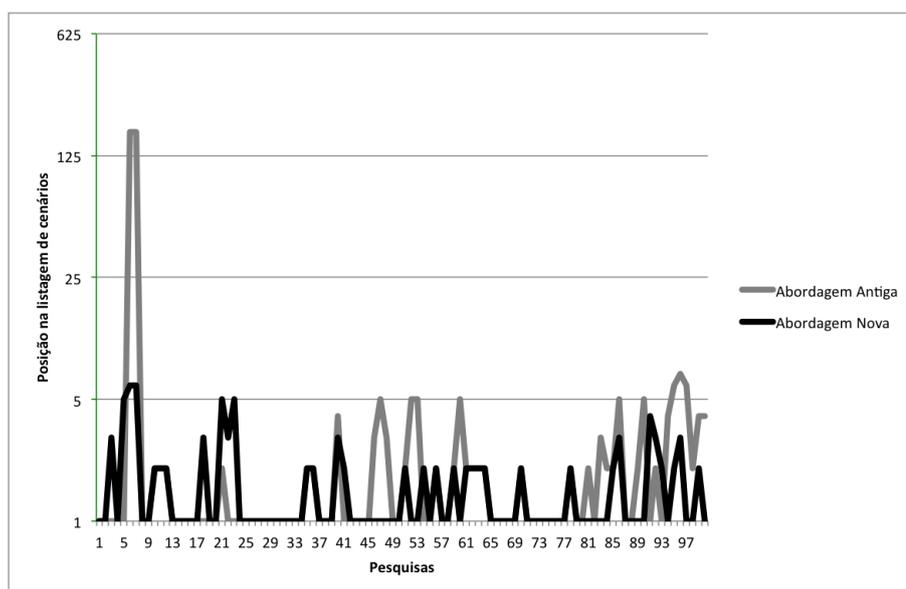


Figura 5.5: Testes quantitativos da ordenação de cenários pelo exercício com utilizadores.

sentadas neste exercício, podendo comparar a média, desvio padrão e o erro associado com intervalo de confiança de 95% de cada abordagem na ordenação de cenários e nas sugestões de *autocomplete*.

Analisando os valores da tabela 5.1, da tabela 5.2 e verificando que o erro associado a um intervalo de confiança de 95% é reduzido pode-se chegar à conclusão que a média diminui quando se compara a abordagem anterior com a nova abordagem, a qual já faz uso do contexto. Pode-se verificar que na ordenação de cenários existe uma diminuição de 0,69 no exercício de contexto simulado e uma diminuição de 3,58 no exercício com utilizadores. Se existe uma diminuição pode-se concluir que a influência do contexto é positiva, pois quando o cenário correto aproxima-se da primeira posição. Esta diferença pode não ser muito grande, mas é o suficiente para desambiguar grande parte dos casos.

Pela análise dos gráficos da figura 5.2 e da figura 5.5 pode-se verificar que a série preta, a qual representa a posição dos cenários fazendo uso do contexto, encontra-se quase sempre

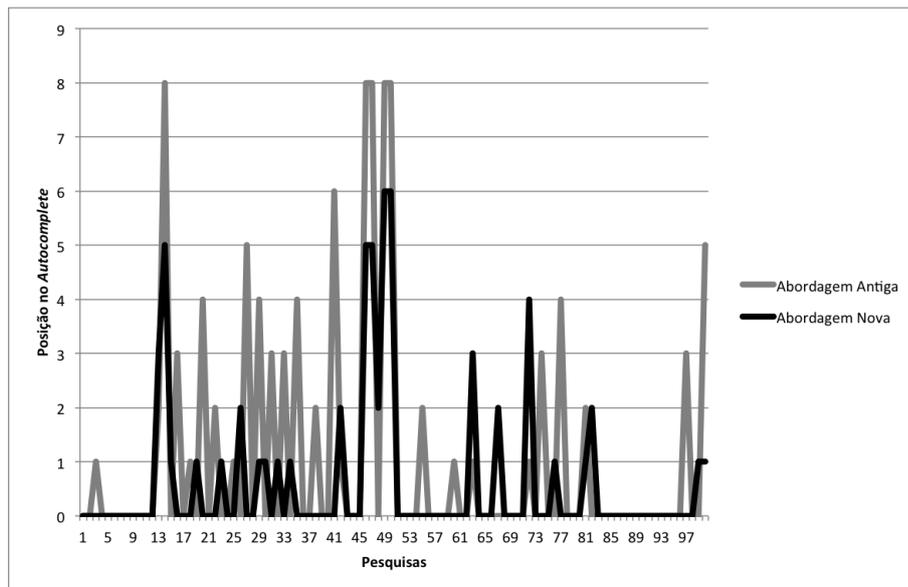


Figura 5.6: Testes quantitativos do *autocomplete* pelo exercício com utilizadores.

Teste	\bar{x}	σ	Erro com Intervalo de Confiança de 95%
Ordenação de Cenários na Abordagem Antiga	5,17	$\pm 23,71$	$\pm 0,15$
Ordenação de Cenários na Abordagem Nova	1,59	$\pm 1,10$	$\pm 0,01$
<i>Autocomplete</i> na Abordagem Antiga	1,04	$\pm 2,09$	$\pm 0,01$
<i>Autocomplete</i> na Abordagem Nova	0,58	$\pm 1,34$	$\pm 0,01$

Tabela 5.2: Estatísticas associadas aos testes realizados nos exercícios com utilizadores.

abaixo da série cinzenta, a qual não faz uso do contexto. Isto significa que a introdução do contexto provocou a subida do cenário correto na listagem de cenários. Na análise destes gráficos pode-se também observar que existem pontos onde a abordagem anterior obteve melhores resultados do que a nova abordagem (intervalo de [16-19] na figura 5.2 e intervalo [20,25] na figura 5.5). Estes pontos podem significar uma mudança de contexto por parte do utilizador, onde os resultados continuam a aparecer consoante o contexto anterior. Nesses casos o utilizador tem que proceder à alteração do cenário e o contexto vai sendo atualizado segundo a interação que faz com o sistema. A partir desses pontos é possível observar que o cenário correto aproxima-se mais da primeira posição usando a nova abordagem.

Passando à análise das sugestões de *autocomplete*. Pelas tabelas 5.1 e 5.2 e verificando que o erro associado a um intervalo de confiança de 95% é reduzido, é possível verificar que a descida da média foi na ordem dos 0,9 no caso do exercício de contexto simulado e 0,46 no caso do exercício com utilizadores. Por estes valores pensa-se que o contexto não tem grande influência. Mas se pensarmos que se está a ter em conta que quando a pesquisa realizada não se encontra nas sugestões de *autocomplete* aparece na oitava posição pode-se chegar à conclusão que as médias apresentadas estão a beneficiar a abordagem anterior, uma vez que apenas nesta abordagem aparecem sugestões a surgir na oitava posição. Esta informação

é visível nos gráficos da figura 5.3 e da figura 5.6. Nestes gráficos é perceptível que existe uma grande oscilação dos valores das posições, mas verifica-se que pela abordagem nova a pesquisa realizada pelo utilizador aproxima-se mais da primeira posição. À semelhança do que ocorre na ordenação de cenários existem pontos onde a abordagem anterior obtém melhores resultados (intervalo [10,11] na figura 5.3 e intervalo [72,74] na figura 5.6), mas isto pode-se dever mais uma vez à mudança de contexto por parte do utilizador.

Nestes testes foram ainda comparados os tempos de geração de cenários usando a abordagem antiga e a abordagem nova. Era de prever que este tempo aumentasse na nova abordagem, uma vez que a complexidade na avaliação de cada cenário aumentou. Pela comparação de tempos visualizada no gráfico da figura 5.4 e na tabela 5.1 verifica-se que essa previsão confirma-se, mas também que essa diferença não é significativa, estando na ordem dos 464,39 ms quando é calculada a diferença entre as suas médias. Na perspetiva da empresa a *trade-off* existente entre o aumento do tempo e o aumento da qualidade é algo que compensa, uma vez que os dados exibidos ao utilizador vão ser mais próximos daqueles que ele está à espera e o tempo aumentado não é o suficiente para o utilizador notar uma grande diferença nas duas abordagens.

5.3.2 Testes Qualitativos

Os testes qualitativos foram realizados recorrendo ao exercício com utilizadores. Além de retirar as métricas já apresentadas anteriormente foi pedido que os utilizadores tecessem alguns comentários e respondessem a algumas questões. De seguida, são apresentadas algumas das respostas dadas pelos utilizadores devidamente tratadas:

1. Notou diferença com a introdução do contexto?
 - "Sim, a ordenação de cenários e o *autocomplete* alteravam consoante as minhas interações com o sistema".
 - "Sim, a desambiguação dos cenários é notória, o que dá bastante jeito uma vez que o número de cenários gerados para uma pesquisa pode ser consideravelmente grande".
2. Percebeu que a mesma pesquisa pode ter interpretações diferentes ao longo da utilização do *Wizdee Discovery*?
 - "Sim, em situações que me tinha esquecido de ver algum pormenor voltava a fazer a mesma pesquisa e a posição do cenário correto já era diferente".
 - "Notório em pesquisas muito ambíguas, onde os mesmos valores aparecem em tabelas diferentes. Se as minhas pesquisas estiverem à volta de uma tabela, o valor apresentado está relacionado com essa tabela e não com outras, nas quais também aparece".
3. O *autocomplete* vem tornar mais fácil a exploração do sistema e adapta-se ao contexto?
 - "Sim, o *autocomplete* ao ordenar as suas sugestões com base nas minhas pesquisas anteriores, permite-me com relativa facilidade aceder às pesquisas que eu desejo realizar de seguida".

4. Comentários?

- "Por vezes, mudo o cenário e a alteração que o contexto introduz não influencia a ordenação de cenários nas pesquisas seguintes e que estão diretamente relacionadas com as anteriores".

Na análise dos testes qualitativos verifica-se que as opiniões são bastante positivas em que os utilizadores referem que a introdução de contexto facilita o uso do sistema e é possível encontrar os resultados que procuram mais rapidamente. Este consegue desambiguar as pesquisas do utilizador em muitos dos casos, fazendo com que o número de alterações de cenários tenha que ser menor. O *autocomplete* consegue ainda ter uma ordenação eficaz das pesquisas mais importantes para o utilizador.

No entanto, pelos comentários recebidos a personalização ainda não consegue proporcionar os melhores resultados ao utilizador. O problema referido prende-se com o facto de as *features* genéricas introduzidas sobrepõem-se às *features* de domínio. Imagine-se uma listagem de cenários, onde o primeiro cenário tem o peso das suas tabelas, colunas e relações entre tabelas a zero e o segundo cenário também. O utilizador ao realizar esta mesma pesquisa procede sempre à alteração de cenário para o segundo, chegando a um ponto que o peso do domínio do segundo cenário atinge o máximo e o cenário mantém-se na segunda posição. Isto acontece, porque os pesos de domínio não são suficientes para conseguirem subir o cenário para primeiro, sendo que os pesos das outras *features* sobrepõem-se.

5.3.3 Análise de Resultados

Ao ser possível a extração das métricas apresentadas através de ambos os exercícios pôde-se analisar a influência do contexto nas funcionalidades de ordenação de cenários e sugestões apresentadas pelo *autocomplete*.

O contexto trouxe resultados muito positivos no sistema, uma vez que o sistema adapta-se ao utilizador através da sua interação. O sistema consegue analisar quais os dados mais importantes para o utilizador e consegue desambiguar muitas das pesquisas introduzidas, levando a que existam casos em que a mesma pesquisa tenha interpretações diferentes em momentos também diferentes.

A introdução do contexto ao ser aplicado na ordenação de cenários possibilita que o cenário que o utilizador deseja apareça numa posição mais acima em relação ao que acontecia na abordagem anterior, a qual era independente do utilizador. Este cenário ao subir na ordenação de cenários proporciona que o utilizador reduza o número de vezes que tem que mudar o cenário e também uma maior rapidez na pesquisa dos dados que são do seu interesse.

Muito semelhante à ordenação de cenários, as sugestões de *autocomplete* também são ordenadas segundo a importância que as diferentes estruturas têm para o utilizador. Desta forma, possibilita que o utilizador execute uma pesquisa mais rapidamente e consiga prever de uma forma mais eficaz qual a pesquisa que o utilizador quer realizar.

No entanto, existem casos que a introdução de contexto não consegue resolver, uma vez que as *features* genéricas sobrepõem-se às *features* de domínio. Desta maneira, é necessário existir trabalho futuro que consiga contrariar esta situação e tornar o sistema mais personalizado para cada utilizador.

Capítulo 6

Metodologia e Planeamento

Este capítulo descreve a metodologia adotada durante o estágio, bem como, o planeamento inicialmente proposto.

6.1 Metodologia

O projeto segue uma metodologia de desenvolvimento denominada de *Scrum* (Rising and Janoff, 2000), uma metodologia iterativa e incremental usada no desenvolvimento de software ágil (Beck et al., 2001), a qual é usada na *Wizdee*. Ao reconhecer que os requisitos de um projeto podem sofrer alterações e que é difícil prever todos os obstáculos que podem surgir durante o desenvolvimento de um projeto de *software*, o *Scrum* procura responder aos desafios que são levantados à medida que estes ocorrem, de uma maneira ágil.

Existem diversas posições no *Scrum*. Em primeiro lugar surge o *Product Owner*, ou Proprietário do Produto, o qual representa o negócio e todas as partes interessadas. De seguida, o *Scrum Master*, ou Mestre do *Scrum*, que tem como função manter a organização da equipa, focando-a nos seus objetivos e protegê-los de destabilizações externas. Por último, surge a Equipa, a qual é responsável pelo desenvolvimento do produto.

O processo de desenvolvimento envolve diversas etapas, denominadas de *Sprints*, os quais têm uma duração normal compreendida entre duas a quatro semanas. As tarefas que são estabelecidas para cada *Sprint*, denominadas de *Sprint Backlog*, são determinadas numa *Sprint Planning Meeting* e são escolhidas a partir do *Product Backlog*, onde todas as tarefas se encontram priorizadas segundo os requisitos a serem cumpridos. Os requisitos que são estabelecidos no *Sprint Backlog* não podem ser alterados, pelo que se não forem cumpridos no tempo estabelecido do *Sprint*, o qual tem que ser sempre terminado a tempo, retornam para o *Product Backlog*.

O progresso do *Sprint* é acompanhado por um *Burn Down Chart*, o qual representa o trabalho total que falta para completar o *Sprint*. Este progresso é ainda discutido diariamente em reuniões com duração aproximadamente de 15 minutos, onde é discutido se o planeamento está a sofrer desvios, quais os objetivos cumpridos e quais os problemas que enfrentam que podem evitar o cumprir do *Sprint*. Tendo este terminado, são realizadas duas reuniões, uma *Sprint Review Meeting*, onde a equipa apresenta o trabalho realizado, e uma *Sprint Retrospective Meeting*, onde é feita uma análise retrospectiva e são realizados melhoramentos no processo.

Na *Wizdee*, as posições referidas acima são representadas pelo Prof. Doutor Paulo Gomes, como *Product Owner*, pelo Doutor Bruno Antunes, como *Scrum Master* e pela Equipa da *Wizdee*, onde o estagiário está inserido. Os *Sprints* têm a duração normal de duas semanas e a reunião diária é realizada entre o *Scrum Master* e a Equipa. No final de cada *Sprint* é realizada a *Sprint Review Meeting*, onde o trabalho realizado é apresentado

por cada elemento da Equipa e a *Sprint Review Meeting* é feita com o objetivo de melhorar o processo adotado. A *Sprint Planning Meeting* é realizada no início de cada *Sprint*.

O processo de *Scrum* está a ser implementado usando o *Jira*¹, o qual oferece uma interface que lida com a gestão das tarefas a serem realizadas no *Sprint*. O código-fonte, bem como qualquer documentação, é armazenada num repositório de versões baseado em *Git* e acessível através da ferramenta *Stash*².

6.2 Planeamento

Numa metodologia de *Scrum*, a criação de um diagrama *Gantt* é algo que não é usual. No entanto, de modo a representar o planeamento usado, surge na figura 6.1 o diagrama respeitante ao primeiro semestre e na figura 6.2 o diagrama respeitante ao segundo semestre. O desenvolvimento do projeto irá ser dividido em doze *sprints*. As reuniões próprias do *Scrum* são realizadas conforme foi referido na metodologia.

As estimativa do tempo das tarefas comuns com outros estágios, como a escrita, é feita com base na experiência que o *Scrum Master* tem na orientação de estágios que já ocorreram na empresa. Nas tarefas de implementação, esta estimativa é realizada de acordo com o tipo de implementação que é realizada e o esforço que esta traduz. Nas tarefas da vertente de engenharia, esta estimativa aproxima-se muito mais da realidade do que nas tarefas da vertente de investigação. Nestas últimas, o processo pode ser mais demorado, levando a que algumas tarefas tenham que ser movidas de um *sprint* para o próximo. Desta forma, e sendo este um risco identificado na realização deste trabalho, é necessário a ativação do plano de mitigação nestas situações.

De seguida irão ser referidos os planeamento do primeiro e segundo semestre em separado.

6.2.1 Primeiro Semestre

O primeiro semestre focou-se na elaboração do presente documento, bem como na implementação da tarefa de Colaboração e Partilha, a qual é descrita na secção Implementação. Na listagem seguinte pode ser visto como foi organizado o planeamento do primeiro semestre, fazendo a correspondência com o diagrama de *Gantt* da figura 6.1:

1. Análise de Requisitos e Riscos - Iniciou-se pela escrita da análise de requisitos (3.3) e riscos (3.6).
2. Análise de Competidores - Exploração das diversas ferramentas de BI e suas respetivas abordagens no campo da colaboração e partilha (2.1.2), e, também, a realização de uma análise de diversos trabalhos relacionados com a área de personalização (2.2.2).
3. Conceitos Base - Especificação das abordagens seguidas tanto a nível de colaboração e partilha (2.1.1), como a nível de personalização (2.2.1).
4. Abordagem Proposta - Foi explorada a arquitetura do *Wizdee Discovery* (3.4) bem como foram explorados os componentes importantes para a realização do estágio (3.5).

¹Jira - <http://www.atlassian.com/software/jira>

²Stash - <http://www.atlassian.com/software/stash>

5. Implementação - Escrita da secção que descreve as funcionalidades implementadas na área da colaboração e partilha (4.1).
6. Proposta de Estágio - Escrita final da proposta de estágio com as alterações propostas pelos orientadores.
7. Desenvolvimento - Em paralelo com a elaboração do documento, foram implementados os métodos necessários da colaboração e partilha.

Um *backlog* foi feito para atribuir as diferentes tarefas a cada *sprint* de desenvolvimento. De seguida é apresentado cada *sprint* ao detalhe com as tarefas respetivas do *backlog*:

- *Sprint* #1:
 - Partilhar relatórios com utilizadores específicos.
 - Identificar relatórios partilhados e mudar as suas propriedades de partilha.
 - Partilha por email
- *Sprint* #2:
 - Partilha por *link*
 - Partilha para redes sociais
- *Sprint* #3:
 - Partilha para ferramentas de produtividade
 - Suporte para comentários
- *Sprint* #4:
 - Suporte para notificações
 - Adicionar suporte para partilha interna de *dashboards*.
 - Implementar método para verificar se a autenticação é válida no acesso à API.

6.2.2 Segundo Semestre

O segundo semestre foi iniciado com a ambientação do estagiário ao algoritmo genético que era usado na abordagem anterior para a otimização da ordenação de cenários. Posteriormente, foram realizadas as necessárias adaptações ao novo motor de pesquisa do *Wizdee Discovery* e foram introduzidas as novas *features*, bem como os respetivos testes.

Quando foram atingidos bons resultados foi iniciada a implementação das funcionalidades de personalização, tornando os resultados devolvidos pela ferramenta mais orientado ao contexto do utilizador. No final desta implementação foram realizados os testes descritos na secção 5.

Por último, e no final do semestre foi escrito o presente documento. Na figura 6.2 é possível ver o diagrama de Gantt respeitante ao segundo semestre.

À semelhança do planeamento do primeiro semestre, um *backlog* foi feito para atribuir as diferentes tarefas a cada *sprint* de desenvolvimento. Muitas das tarefas passam de um *sprint* para o outro devido à sua complexidade e tempo consumido. De seguida é apresentado cada *sprint* ao detalhe com as tarefas respetivas do *backlog*:

- *Sprint #5:*
 - Melhorar ordenação de cenários através da abordagem usada.
- *Sprint #6:*
 - Melhorar ordenação de cenários através da abordagem usada.
 - Testes de validação das novas *features*.
- *Sprint #7:*
 - Melhorar ordenação de cenários através da abordagem usada.
 - Testes de validação das novas *features*.
 - Não mostrar notificações relacionadas com *dashboards* e *widgets* que já não existem.
- *Sprint #8:*
 - Implementar um *Context-Manager* e métodos necessários.
- *Sprint #9:*
 - Implementar um *Context-Manager* e métodos necessários.
- *Sprint #10:*
 - Implementar um *Context-Manager* e métodos necessários.
 - Aplicação do *Context-Manager* ao método de avaliação de cenários.
- *Sprint #11:*
 - Aplicação do *Context-Manager* às sugestões do *autocomplete*.
 - Aplicação do *Context-Manager* às pesquisas sugeridas
- *Sprint #12:*
 - Testes de validação da personalização.

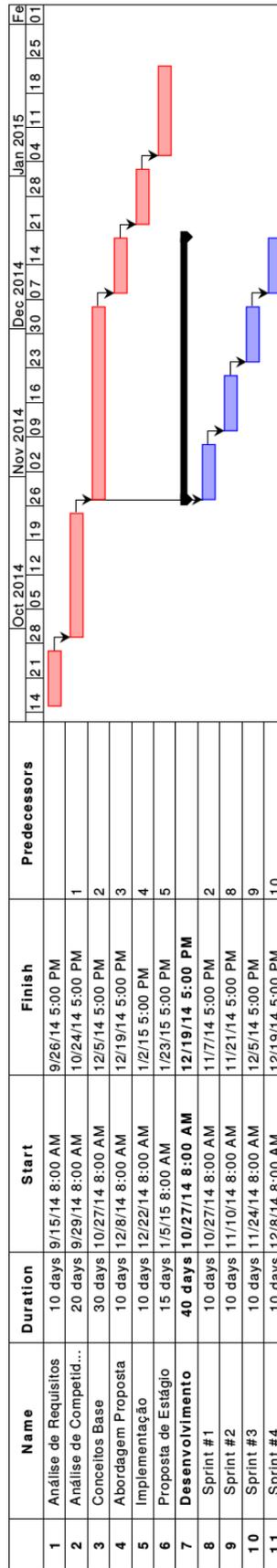


Figura 6.1: Diagrama de Gantt respeitante ao primeiro semestre.

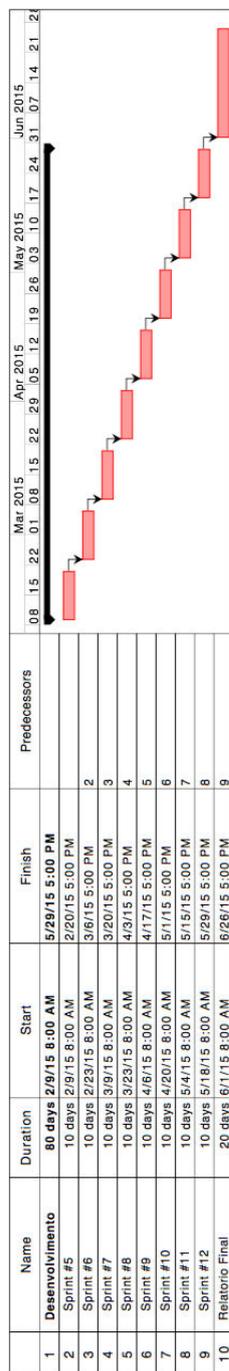


Figura 6.2: Diagrama de Gantt respeitante ao segundo semestre.

Capítulo 7

Conclusões

O trabalho desenvolvido durante este estágio complementa o *Wizdee Discovery* na área da partilha de informação e da personalização.

7.1 Objetivos

As implementações realizadas contribuem para o melhoramento da experiência do utilizador na interação com o sistema. Uma delas foca-se mais na colaboração, a qual permite a partilha de relatórios tanto entre utilizadores como para o exterior, como redes sociais ou ferramentas de produtividade. Num sistema de BI, este tipo de funcionalidades torna-se importante uma vez que os vários departamentos de uma empresa necessitam de estar em permanente comunicação, possibilitando uma eficiente análise dos dados e a troca de informação e ideias. Por outro lado, a personalização do sistema tenta adaptar as respostas ao utilizador. Através das interações que este realiza com o sistema é possível a modelização de um contexto do utilizador, capaz de influenciar as pesquisas sugeridas bem como as interpretações que são realizadas pelo sistema quando este realiza uma pesquisa. Com esta implementação pretende-se que o *Wizdee Discovery* adequa as respostas apresentadas às necessidades do utilizador, o qual pode resolver problemas de ambiguidade, os quais são um dos maiores problemas do processamento de linguagem natural.

7.2 Abordagem

De maneira a entender o que pode ser implementado na área da colaboração, foi importante a análise de competidores realizada, onde foi possível o levantamento das abordagens que estes seguiam nos seus sistemas, presente no capítulo *Conceitos Base e Trabalhos Relacionados*. Este estudo deu para perceber quais as abordagens usadas pelos concorrentes da *Wizdee* e onde poderia o *Wizdee Discovery* ser diferente. A implementação da partilha para ferramentas de produtividade foi claramente diferenciador, pois nenhuma das ferramentas de BI analisadas permite este tipo de partilha. Ainda neste capítulo foram estudadas as abordagens que diversos investigadores usam na personalização de motores de pesquisa. Desta maneira permitiu ter uma visão geral de como este tipo de aprendizagem pode ser feito e algumas das considerações a ter. Uma das considerações mais importantes prende-se com o facto de que a personalização não deve ser feita a todo o tipo de pesquisas. Outra realça que o contexto de utilizador pode sofrer muitas alterações. Estes são alguns dos problemas que foram tidos em conta no momento da implementação da personalização.

Outro dos aspetos importantes, foi a definição dos requisitos e objetivos do estágio, os quais permitiram a sincronização das ideias entre o estagiário e a empresa. O estudo da

arquitetura permitiu identificar o contexto onde o trabalho iria ser desenvolvido e quais as dependências existentes entre os diversos projetos. A análise de riscos foi também importante, na medida em que na existência de uma situação de risco prevista, um plano de contingência está desde já identificado.

Desta forma, foram usadas as seguintes abordagens:

- **Colaboração e Partilha.** Para a colaboração e partilha foram usadas algumas das abordagens já seguidas por alguns dos competidores da *Wizdee*. Assim e através da gestão da base de dados e alteração dos respetivos comandos de controlo, foi possível a implementação da partilha interna, partilha por *link*, suporte para comentários e suporte para notificações. Já para a implementação da partilha externa, foram usadas, na maioria dos casos, bibliotecas externas que permitem a autenticação e partilha por email, redes sociais e ferramentas de produtividade. Nos outros casos em que as bibliotecas eram pagas ou incapazes de realizar o que se queria, os métodos para aceder à API foram implementados de raiz.
- **Ordenação de Cenários.** Para a ordenação de cenários foram usados algoritmos genéticos capazes de calcular o peso associado a cada uma das *features* introduzidas. Estas *features* têm a função de avaliar cada cenário e podem ser divididas em *features* genéricas, as quais são independentes do domínio dos dados, e em *features* de domínio. É sobre estas últimas que a personalização atua.
- **Personalização.** A personalização atua sobre as *features* de domínio. Estas baseiam-se nos pesos atribuídos a cada estrutura interpretada nos cenários, como tabelas, colunas e relações, e são as interações do utilizador com o sistema que fazem variar estes pesos, segundo a importância que cada estrutura tem para o utilizador.

7.3 Resultados

Os resultados obtidos revelaram que as funcionalidades introduzidas trouxeram benefícios para o *Wizdee Discovery*. Nos seguintes parágrafos irão ser analisados os resultados para cada uma das áreas tratadas neste estágio.

- **Colaboração e Partilha.** Pelos testes unitários realizados e testes de aceitação, as funcionalidades de colaboração e partilha foram bem integradas no sistema e satisfazem os requisitos propostos.
- **Ordenação de Cenários.** Comparando a ordenação dos cenários entre a abordagem anterior e a abordagem nova conclui-se que os resultados são positivos, levando a um melhoramento significativo na capacidade para acertar no cenário correto. A complexidade aumentada no método de avaliação de cada cenário levou que o tempo de avaliação aumentasse, o que era de esperar. No entanto, não revelou ser um aumento significativo para o utilizador, o qual nunca ultrapassou os 500 ms na diferença entre a geração dos mesmos cenários usando a abordagem anterior e usando a abordagem nova.
- **Personalização.** A personalização revelou-se prometedora, uma vez que foi capaz de desambiguar muitas das pesquisas introduzidas pelo utilizador e adaptar os resultados a diferentes utilizador através das suas interações. No entanto, existem casos que o contexto não é capaz de influenciar as ordenações de cenários e *autocomplete*. Este é um problema que deve ser tido em conta no trabalho futuro.

7.4 Contribuições

O estagiário durante este estágio contribuiu de uma forma positiva nas seguintes áreas:

- **Colaboração e Partilha.** A implementação das funcionalidades de colaboração e partilha possibilitam o utilizador de partilhar internamente os seus relatórios, enviá-los por email, partilhar para redes sociais os seus *widgets* e também interagir com algumas das suas ferramentas de produtividade. O suporte para comentários também foi implementado, podendo agora os utilizadores comentar os seus próprios livros, páginas e *widgets* bem como os de outros utilizadores. Nestes comentários é ainda possível o uso de menções que irão fazer referência a outros utilizadores ou relatórios. De maneira a que os utilizadores percebam que algo foi partilhado com eles ou foram mencionados num comentário ou então foi escrito um comentário num relatório seu, um sistema de notificações foi também introduzido no sistema, avisando o utilizador destas ações. Importante realçar que estas funcionalidades foram implementadas a nível de *back-end*, estando a parte de *front-end* da responsabilidade da equipa existente na *Wizdee* dedicada ao desenvolvimento do interface de utilizador.
- **Ordenação de Cenários.** O método de ordenação de cenários foi enriquecido com diversas *features* que permitem uma avaliação mais complexa e mais fiável de cada cenário. Desta forma, é possível neste momento uma melhor interpretação de cada pesquisa introduzida pelo utilizador, dando na maioria dos casos uma resposta correta ao utilizador.
- **Personalização.** Apesar da melhoria significativa na ordenação de cenários, existem casos em que falha devido à ambiguidade das pesquisas introduzidas pelo utilizador. Desta maneira, o sistema encontra-se personalizado para cada utilizador, sendo que os resultados são adaptados para o utilizador de acordo com a sua importância para ele. Esta personalização é realizada através da interação do utilizador com o sistema, não tendo que o próprio utilizador dar o seu *input* para dizer quais os dados que deseja ver. Esta funcionalidade foi aplicada ao *autocomplete*, ordenação de cenários e pesquisas sugeridas.

7.5 Trabalho Futuro

Como trabalho futuro, a nível da colaboração e partilha podem ser implementadas mais conectores para outras redes sociais e ferramentas de produtividade, conforme as necessidades dos clientes e seu *feedback*.

Por outro lado, a nível da personalização, o processo deve ser melhorado. Existem casos em que o contexto é incapaz de ter influência na ordenação de cenários e nas sugestões de pesquisa apresentados, uma vez que as *features* genéricas têm mais peso que as *features* de domínio. Uma solução possível para resolver estes casos é a atribuição de um peso a cada conjunto de *features*, sendo estes também personalizáveis consoante as interações do utilizador com o sistema. Diminuindo o peso das *features* genéricas e aumentando o peso das *features* de domínio, é possível que os cenários se ordenem segundo os interesses do utilizador sem estarem tão dependentes das *features* genéricas.

Apêndice A

Referência do API de Colaboração

Tabela A.1: Referência do API de Colaboração.

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.01	<i>Add Book</i>	Adiciona um livro onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador do utilizador; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<ul style="list-style-type: none"> - Identificador do livro; - Identificador do utilizador; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha.
Ep.02	<i>Edit Book</i>	Edita um livro onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador do livro; - Identificador do utilizador; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<i>Boolean</i>
Ep.03	<i>Remove Book</i>	Remove um livro bem como todos os registos relativos à partilha deste.	Identificador do Livro	<i>Boolean</i>
Ep.04	<i>Get Books</i>	Obtém os livros. Segundo um parâmetro booleano pode ser definido se queremos apenas os partilhados, os não partilhados ou todos.	<ul style="list-style-type: none"> - Parâmetro booleano a definir quais os livros a serem visualizados. 	Lista de livros
Continua na próxima página				

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.05	<i>Add BookPage</i>	Adiciona uma página onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador do utilizador; - Identificador do livro; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<ul style="list-style-type: none"> - Identificador da página; - Identificador do utilizador; - Identificador do livro; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha.
Ep.06	<i>Edit BookPage</i>	Edita uma página onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador da página; - Identificador do utilizador; - Identificador do livro; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<ul style="list-style-type: none"> - Identificador da página; - Identificador do utilizador; - Identificador do livro; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha.
Ep.07	<i>Remove BookPage</i>	Remove uma página bem como todas os registos relativos à partilha desta.	Identificador da Página	<i>Boolean</i>
Ep.08	<i>Get BookPages</i>	Obtém as páginas.	<ul style="list-style-type: none"> - Identificador do utilizador; - Identificador do livro a que pertence a página. 	Lista de páginas

Continua na próxima página

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.09	<i>Add Widget</i>	Adiciona um <i>widget</i> onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador do utilizador; - Identificador da página; - Identificador do livro; - Título; - Identificadores dos utilizados com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<ul style="list-style-type: none"> - Identificador do <i>widget</i> - Identificador do utilizador; - Identificador da página; - Identificador do livro; - Título; - Identificadores dos utilizados com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha.
Ep.10	<i>Edit Widget</i>	Edita um <i>widget</i> onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador do <i>widget</i> - Identificador do utilizador; - Identificador da página; - Identificador do livro; - Título; - Identificadores dos utilizados com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<ul style="list-style-type: none"> - Identificador do <i>widget</i> - Identificador do utilizador; - Identificador da página; - Identificador do livro; - Título; - Identificadores dos utilizados com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha.
Ep.11	<i>Remove Widget</i>	Remove um <i>widget</i> bem como todos os registos relativos à partilha deste.	Identificador do <i>Widget</i>	<i>Boolean</i>
Ep.12	<i>Get Widgets</i>	Obtém os <i>widgets</i>	<ul style="list-style-type: none"> - Identificador do utilizador; - Identificador do livro a que pertence o <i>widget</i>; - Identificador da página a que pertence o <i>widget</i>. 	Lista de <i>widgets</i>
Continua na próxima página				

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.13	<i>Add Dashboard</i>	Adiciona um <i>dashboard</i> onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador do utilizador; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<ul style="list-style-type: none"> - Identificador do <i>dashboard</i>; - Identificador do utilizador; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha.
Ep.14	<i>Edit Dashboard</i>	Edita um <i>dashboard</i> onde podem ser definidos os utilizadores e grupos com quem é feita a partilha.	<ul style="list-style-type: none"> - Identificador do <i>dashboard</i>; - Identificador do utilizador; - Título; - Identificadores dos utilizadores com quem é feita a partilha; - Identificadores dos grupos com quem é feita a partilha. 	<i>Boolean</i>
Ep.15	<i>Remove Dashboard</i>	Remove um <i>dashboard</i> bem como todos os registos relativos à partilha deste.	Identificador do <i>dashboard</i>	<i>Boolean</i>
Ep.16	<i>Get Dashboards</i>	Obtém os <i>dashboards</i> . Segundo um parâmetro booleano pode ser definido se queremos apenas os partilhados, os não partilhados ou todos.	<ul style="list-style-type: none"> - Parâmetro booleano a definir quais os <i>dashboards</i> a serem visualizados. 	Lista de <i>dashboards</i>
Continua na próxima página				

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.17	<i>Add Share Link</i>	Cria um <i>link</i> para partilha	<ul style="list-style-type: none"> - Duração; - Identificador do objeto a ser partilhado; - Permissões. 	URL gerado
Ep.18	<i>Edit Share Link</i>	Edita as propriedades de um <i>link</i> para partilha	<ul style="list-style-type: none"> - Identificador do <i>link</i> partilhado - Duração; - Identificador do objeto a ser partilhado; - Permissões; - URL. 	<i>Boolean</i>
Ep.19	<i>Remove Share Link</i>	Remove o <i>link</i> para partilha	Identificador do <i>Link</i> Partilhado	<i>Boolean</i>
Ep.20	<i>Send Email</i>	Envia um email com o <i>widget</i> desejado em anexo	<ul style="list-style-type: none"> - Email do emissor; - Email(s) do(s) destinatário(s); - Assunto; - Texto; - Propriedades do <i>Widget</i> 	<i>Boolean</i>
Ep.21	<i>Share Facebook</i>	Partilha um <i>widget</i> no <i>Facebook</i> como imagem com a descrição à escolha do utilizador.	<ul style="list-style-type: none"> - Identificador do utilizador - Identificador do álbum - Descrição - URL da Imagem. 	<i>Boolean</i>
Ep.22	<i>Generate Access Token Facebook</i>	Gera um <i>access token</i> a partir de um <i>request token</i> .	<ul style="list-style-type: none"> - Identificador do utilizador; - <i>Request token</i>. 	<i>Access Token</i>
Ep.23	<i>Create Album Facebook</i>	Cria um novo álbum no <i>Facebook</i>	<ul style="list-style-type: none"> - Identificador do utilizador; - <i>Nome do álbum</i>. 	<i>Boolean</i>
Ep.24	<i>Get Albums Facebook</i>	Obtém os álbuns do <i>Facebook</i> que o utilizador possui.	Identificador do utilizador.	Lista com o nome dos álbuns.
Continua na próxima página				

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.25	<i>Check Access Token Facebook</i>	Verifica a validade do <i>access token</i> que se encontra na base de dados.	Identificador do utilizador.	<i>Boolean</i>
Ep.26	<i>Share Twitter</i>	Partilha um <i>widget</i> no <i>Twitter</i> como imagem com a descrição à escolha do utilizador.	- Identificador do utilizador - Descrição - URL da Imagem.	<i>Boolean</i>
Ep.27	<i>Generate Access Token Twitter</i>	Gera um <i>access token</i> a partir de um <i>request token</i> .	- Identificador do utilizador; - <i>Request token</i> .	<i>Access Token</i>
Ep.28	<i>Check Access Token Twitter</i>	Verifica a validade do <i>access token</i> que se encontra na base de dados.	Identificador do utilizador.	<i>Boolean</i>
Ep.29	<i>Share LinkedIn</i>	Partilha um <i>widget</i> no <i>LinkedIn</i> como imagem com a descrição à escolha do utilizador.	- Identificador do utilizador - Descrição - URL da Imagem.	<i>Boolean</i>
Ep.30	<i>Generate Access Token LinkedIn</i>	Gera um <i>access token</i> a partir de um <i>request token</i> .	- Identificador do utilizador; - <i>Request token</i> .	<i>Access Token</i>
Ep.31	<i>Check Access Token LinkedIn</i>	Verifica a validade do <i>access token</i> que se encontra na base de dados.	Identificador do utilizador.	<i>Boolean</i>
Ep.32	<i>Share Dropbox</i>	Adiciona livros, páginas e <i>widgets</i> na conta da <i>Dropbox</i> do utilizador.	- Identificador do utilizador - Identificador do objeto - URL do objeto.	<i>Boolean</i>
Ep.33	<i>Generate Access Token Dropbox</i>	Gera um <i>access token</i> a partir de um <i>request token</i> .	- Identificador do utilizador; - <i>Request token</i> .	<i>Access Token</i>
Ep.34	<i>Check Access Token Dropbox</i>	Verifica a validade do <i>access token</i> que se encontra na base de dados.	Identificador do utilizador.	<i>Boolean</i>
Ep.35	<i>Create Notebook Evernote</i>	Cria um novo bloco de notas no <i>Evernote</i> na conta do utilizador.	- Identificador do utilizador - Nome do bloco de notas.	<i>Boolean</i>
Ep.36	<i>Get Notebooks Evernote</i>	Obtém os blocos de notas do <i>Evernote</i> que o utilizador possui.	Identificador do utilizador.	Lista com os nomes dos blocos de notas.
Continua na próxima página				

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.37	<i>Add Note Evernote</i>	Adicionar uma nota no bloco de notas que o utilizador escolheu na sua conta <i>Evernote</i>	<ul style="list-style-type: none"> - Identificador do utilizador - Identificador do bloco de notas - Título da nota - Descrição - URL do objeto; - Formato do anexo. 	<i>Boolean</i>
Ep.38	<i>Generate Access Token Evernote</i>	Gera um <i>access token</i> a partir de um <i>request token</i> .	<ul style="list-style-type: none"> - Identificador do utilizador; - <i>Request token</i>. 	<i>Access Token</i>
Ep.39	<i>Check Access Token Evernote</i>	Verifica a validade do <i>access token</i> que se encontra na base de dados.	Identificador do utilizador.	<i>Boolean</i>
Ep.40	<i>Add Event to Google Calendar</i>	Adiciona um evento ao Google Calendar associado.	<ul style="list-style-type: none"> - Identificador do utilizador; - Data do início do evento; - Data do fim do evento; - Sumário; - Localização; - Descrição; - Emails dos convidados; - Parâmetro booleano para definir se o evento pode ser criado sobre outro; - URL do objeto; - Formato do anexo. 	<i>Boolean</i>
Ep.41	<i>Generate Access Token Google Calendar</i>	Gera um <i>access token</i> a partir de um <i>request token</i> .	<ul style="list-style-type: none"> - Identificador do utilizador; - <i>Request token</i>. 	<i>Access Token</i>
Ep.42	<i>Check Access Token Google Calendar</i>	Verifica a validade do <i>access token</i> que se encontra na base de dados.	Identificador do utilizador.	<i>Boolean</i>

Continua na próxima página

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.43	<i>Add Board to Trello</i>	Adiciona quadros na conta <i>Trello</i> do utilizador.	- Identificador do utilizador - Nome do quadro.	Identificador do Quadro
Ep.44	<i>Get Boards Trello</i>	Obtém os quadros da conta <i>Trello</i> do utilizador.	Identificador do utilizador.	Lista de nomes dos quadros com respetivos identificadores.
Ep.45	<i>Add Card to Trello</i>	Adiciona cartões na conta <i>Trello</i> do utilizador de dada lista e quadro.	- Identificador do utilizador; - Identificador da lista; - Nome do cartão; - Descrição; - Data limite; - URL do objeto.	Identificador do Cartão
Ep.46	<i>Get Cards Trello</i>	Obtém os cartões da conta <i>Trello</i> do utilizador de dada lista.	- Identificador do utilizador - Identificador da lista.	Lista de nomes dos cartões com respetivos identificadores
Ep.47	<i>Add List to Trello</i>	Adiciona listas na conta <i>Trello</i> do utilizador de dado quadro.		Identificador da Lista
Ep.48	<i>Get Lists Trello</i>	Obtém as listas da conta <i>Trello</i> do utilizador de dado quadro.	- Identificador do utilizador - Identificador do quadro.	Lista de listas com respetivos identificadores
Ep.49	<i>Generate Access Token Trello</i>	Gera um <i>access token</i> a partir de um <i>request token</i> .	- Identificador do utilizador; - <i>Request token</i> .	<i>Access Token</i>
Ep.50	<i>Check Access Token Trello</i>	Verifica a validade do <i>access token</i> que se encontra na base de dados.	Identificador do utilizador.	<i>Boolean</i>
Ep.51	<i>Add Comment</i>	Adiciona comentários a determinado livro, página ou <i>widget</i> . Permite a realização de menções.	-Texto; - Identificador do utilizador que escreveu; - Identificador do utilizador que é proprietário do objeto; - Identificador do objeto.	- Identificador do comentário - Texto; - Identificador do utilizador que escreveu; - Identificador do utilizador que é proprietário do objeto; - Identificador do objeto.

Continua na próxima página

ID	Nome	Descrição	Dados de Entrada	Dados de Saída
Ep.52	<i>Edit Comment</i>	Edita um comentário previamente realizado.	<ul style="list-style-type: none"> - Identificador do comentário - Texto; - Identificador do utilizador que escreveu; - Identificador do utilizador que é proprietário do objeto; - Identificador do objeto. 	<i>Boolean</i>
Ep.53	<i>Remove Comments</i>	Remove um comentário.	Identificador do comentário.	<i>Boolean</i>
Ep.54	<i>Get Comments</i>	Obtém os comentários de dado livro, página ou <i>widget</i> .	Identificador do objeto.	Lista de comentários
Ep.55	<i>Get Activities</i>	Obtém as notificações de dado utilizador.	Identificador do utilizador.	Lista de atividades
Ep.56	<i>Edit Activity as seen</i>	Marca as atividades como vistas para dado utilizador.	Identificador da atividade	<i>Boolean</i>
Ep.57	<i>Edit Activity Settings</i>	Edita as definições relativas às notificações que um utilizador deseja receber.	<ul style="list-style-type: none"> - Identificador do utilizador; - Chave; - Valor 	<i>Boolean</i>
Ep.58	<i>Get Activity Settings</i>	Obtém as definições relativas às notificações de dado utilizador.	Identificador do utilizador.	Listagem das definições do utilizador

Apêndice B

Testes Unitários de Colaboração e Partilha

Os testes unitários que foram usados para a validação das funcionalidades de colaboração e partilha foram os seguintes:

USER STORY: PARTILHA-INTERNA-01

CENÁRIO: PARTILHA-INTERNA-01-SCE-01:

Dado que sou um utilizador

Quando partilho um *widget* com outro utilizador

Então verifico que esse utilizador tem acesso ao meu *widget*

CENÁRIO: PARTILHA-INTERNA-01-SCE-02:

Dado que sou um utilizador

Quando partilho um *widget* com um grupo

Então verifico que todos os utilizadores desse grupo têm acesso ao meu *widget*

CENÁRIO: PARTILHA-INTERNA-01-SCE-03:

Dado que sou um utilizador

Quando partilho um *dashboard* com outro utilizador

Então verifico que esse utilizador tem acesso ao meu *dashboard*, bem como todos os *widgets* pertencentes a esse *dashboard*

CENÁRIO: PARTILHA-INTERNA-01-SCE-04:

Dado que sou um utilizador

Quando partilho um *dashboard* com um grupo

Então verifico que todos os utilizadores desse grupo têm acesso ao meu *dashboard*, bem como todos os *widgets* pertencentes a esse *dashboard*

CENÁRIO: PARTILHA-INTERNA-01-SCE-05:

Dado que sou um utilizador

Quando partilho um *dashboard* publicamente

Então verifico que todos os utilizadores têm acesso ao meu *dashboard*, bem como todos os *widgets* pertencentes a esse *dashboard*

USER STORY: PARTILHA-EXTERNA-01

CENÁRIO: PARTILHA-EXTERNA-01-SCE-01:

Dado que sou um utilizador

Quando gero um *link*

Então verifico que os utilizadores que têm acesso ao *link* conseguem ver a informação pretendida

CENÁRIO: PARTILHA-EXTERNA-01-SCE-02:

Dado que sou um utilizador

Quando associo um tempo de validade ao *link*

Então o *link* não é mais acessível após o término do tempo de validade

CENÁRIO: PARTILHA-EXTERNA-01-SCE-03:

Dado que sou um utilizador

Quando revogo um *link*

Então o *link* não é mais acessível

USER STORY: PARTILHA-EXTERNA-02

CENÁRIO: PARTILHA-EXTERNA-02-SCE-01:

Dado que sou um utilizador

Quando partilho um *widget* por email em formato PDF

Então verifico que os utilizadores para quem eu envio o email recebem esse *widget* em formato PDF

CENÁRIO: PARTILHA-EXTERNA-02-SCE-02:

Dado que sou um utilizador

Quando partilho um *widget* por email em formato PNG

Então verifico que os utilizadores para quem eu envio o email recebem esse *widget* em formato PNG

CENÁRIO: PARTILHA-EXTERNA-02-SCE-03:

Dado que sou um utilizador

Quando partilho um *widget* por email em formato SVG

Então verifico que os utilizadores para quem eu envio o email recebem esse *widget* em formato SVG

USER STORY: PARTILHA-EXTERNA-03

CENÁRIO: PARTILHA-EXTERNA-03-SCE-01:

Dado que sou um utilizador

Quando tento associar a minha conta do *Facebook*

Então verifico que sou reencaminhado para a página do *Facebook* para permitir o *Wizdee Discovery* a partilhar os meus *widgets*

CENÁRIO: PARTILHA-EXTERNA-03-SCE-02:

Dado que sou um utilizador

Quando crio um novo álbum no *Facebook* através do *Wizdee Discovery*

Então verifico que o álbum é criado no meu *Facebook*

CENÁRIO: PARTILHA-EXTERNA-03-SCE-03:

Dado que sou um utilizador

Quando partilho um *widget* para o *Facebook* no álbum que eu especifiquei

Então verifico que o meu *widget* aparece no *Facebook* no álbum que especifiquei

CENÁRIO: PARTILHA-EXTERNA-03-SCE-04:

Dado que sou um utilizador

Quando tento associar a minha conta do *Twitter*

Então verifico que sou reencaminhado para a página do *Twitter* para permitir o *Wizdee Discovery* a partilhar os meus *widgets*

CENÁRIO: PARTILHA-EXTERNA-03-SCE-05:

Dado que sou um utilizador

Quando partilho um *widget* para o *Twitter* com a descrição que especifiquei

Então verifico que o meu *widget* aparece no *Twitter* com a descrição que especifiquei

CENÁRIO: PARTILHA-EXTERNA-03-SCE-06:

Dado que sou um utilizador

Quando tento associar a minha conta do *LinkedIn*

Então verifico que sou reencaminhado para a página do *LinkedIn* para permitir o *Wizdee Discovery* a partilhar os meus *widgets*

CENÁRIO: PARTILHA-EXTERNA-03-SCE-07:

Dado que sou um utilizador

Quando partilho um *widget* para o *LinkedIn* com a descrição que especifiquei

Então verifico que o meu *widget* aparece no *LinkedIn* com a descrição que especifiquei

USER STORY: PARTILHA-EXTERNA-04

CENÁRIO: PARTILHA-EXTERNA-04-SCE-01:

Dado que sou um utilizador

Quando tento associar a minha conta da *Dropbox*

Então verifico que sou reencaminhado para a página da *Dropbox* para permitir o *Wizdee Discovery* a partilhar a minha informação

CENÁRIO: PARTILHA-EXTERNA-04-SCE-02:

Dado que sou um utilizador

Quando tento partilhar um livro para a *Dropbox*

Então verifico que o meu livro aparece na minha *Dropbox*

CENÁRIO: PARTILHA-EXTERNA-04-SCE-03:

Dado que sou um utilizador

Quando tento partilhar uma página para a *Dropbox*

Então verifico que a minha página aparece na minha *Dropbox*

CENÁRIO: PARTILHA-EXTERNA-04-SCE-04:

Dado que sou um utilizador

Quando tento partilhar um *widget* para a *Dropbox*

Então verifico que o meu *widget* aparece na minha *Dropbox*

CENÁRIO: PARTILHA-EXTERNA-04-SCE-05:

Dado que sou um utilizador

Quando tento associar a minha conta da *Evernote*

Então verifico que sou reencaminhado para a página da *Evernote* para permitir o *Wizdee Discovery* a partilhar a minha informação

CENÁRIO: PARTILHA-EXTERNA-04-SCE-06:

Dado que sou um utilizador

Quando tento criar uma nota no *Evernote*

Então verifico que a minha nota aparece no meu *Evernote*

CENÁRIO: PARTILHA-EXTERNA-04-SCE-07:

Dado que sou um utilizador

Quando tento criar uma nota no *Evernote* com um objeto associado

Então verifico que a minha nota aparece no meu *Evernote* com o objeto associado

CENÁRIO: PARTILHA-EXTERNA-04-SCE-08:

Dado que sou um utilizador

Quando tento associar a minha conta da *Google Calendar*

Então verifico que sou reencaminhado para a página da *Google* para permitir o *Wizdee Discovery* a criar eventos no calendário

CENÁRIO: PARTILHA-EXTERNA-04-SCE-09:

Dado que sou um utilizador

Quando tento criar um evento no *Google Calendar* com as especificações que eu dou

Então verifico que o meu evento é criado no meu *Google Calendar* com essas mesmas especificações

CENÁRIO: PARTILHA-EXTERNA-04-SCE-10:

Dado que sou um utilizador

Quando tento associar a minha conta do *Trello*

Então verifico que sou reencaminhado para a página do *Trello* para permitir o *Wizdee Discovery* a partilhar a minha informação

CENÁRIO: PARTILHA-EXTERNA-04-SCE-11:

Dado que sou um utilizador

Quando tento adicionar um quadro no *Trello*

Então verifico que o meu quadro aparece no meu *Trello*

CENÁRIO: PARTILHA-EXTERNA-04-SCE-12:

Dado que sou um utilizador

Quando tento adicionar um cartão no *Trello*

Então verifico que o meu cartão aparece no meu *Trello*

CENÁRIO: PARTILHA-EXTERNA-04-SCE-13:

Dado que sou um utilizador

Quando tento adicionar uma lista no *Trello*

Então verifico que a minha lista aparece no meu *Trello*

USER STORY: COMENTÁRIOS-01

CENÁRIO: COMENTÁRIOS-01-SCE-01:

Dado que sou um utilizador
Quando tento adicionar um comentário a um livro
Então verifico que o meu comentário aparece nesse livro

CENÁRIO: COMENTÁRIOS-01-SCE-02:

Dado que sou um utilizador
Quando tento adicionar um comentário a uma página
Então verifico que o meu comentário aparece nessa página

CENÁRIO: COMENTÁRIOS-01-SCE-03:

Dado que sou um utilizador
Quando tento adicionar um comentário a um *widget*
Então verifico que o meu comentário aparece nesse *widget*

CENÁRIO: COMENTÁRIOS-01-SCE-04:

Dado que sou um utilizador
Quando tento editar um comentário
Então verifico que o meu comentário é editado

CENÁRIO: COMENTÁRIOS-01-SCE-05:

Dado que sou um utilizador
Quando tento remover um comentário
Então verifico que o meu comentário é removido

CENÁRIO: COMENTÁRIOS-01-SCE-06:

Dado que sou um utilizador
Quando tento obter os comentários de um determinado objeto
Então verifico que obtenho os comentários

USER STORY: NOTIFICAÇÕES-01

CENÁRIO: NOTIFICAÇÕES-01-SCE-01:

Dado que sou um utilizador
Quando tento obter as minhas atividades
Então verifico que obtenho as minhas atividades

CENÁRIO: NOTIFICAÇÕES-01-SCE-02:

Dado que sou um utilizador
Quando vejo uma atividade
Então verifico que essa atividade é marcada como vista

CENÁRIO: NOTIFICAÇÕES-01-SCE-03:

Dado que sou um utilizador
Quando edito as definições de atividade
Então verifico que essas definições são corretamente aplicadas

CENÁRIO: NOTIFICAÇÕES-01-SCE-04:

Dado que sou um utilizador
Quando peço as minhas definições de atividade
Então verifico que as minhas definições são corretamente exibidas

Bibliografia

- Adomavicius, G. and Tuzhilin, A. (2011). Context-Aware Recommender Systems. In *Recommender Systems Handbook*, pages 217–253. Springer US.
- Aioli, F. and Sperduti, A. (2010). A preference optimization based unifying framework for supervised learning problems. In Fürnkranz, J. and Hüllermeier, E., editors, *Preference Learning*, pages 19–42. Springer-Verlag.
- Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, 2nd edition.
- Anand, S. and Mobasher, B. (2007). Contextual Recommendation. In *From Web to Social Web: Discovering and Deploying User and Content Profiles*, volume 4737 of *Lecture Notes in Computer Science*, pages 142–160. Springer-Verlag, Berlin, Heidelberg.
- Anand, S. S. and Mobasher, B. (2005). Intelligent techniques for web personalization. In *Intelligent Techniques for Web Personalization*, volume 3169 of *Lecture Notes in Computer Science*, pages 1–36. Springer Berlin / Heidelberg.
- Antoniou, G. and Harmelen, F. v. (2008). *A Semantic Web Primer, 2Nd Edition (Cooperative Information Systems)*. The MIT Press, 2 edition.
- Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In *Proc. of the 15th National Conference on Artificial Intelligence (AAAI '98/IAAI '98)*, pages 714–720, Menlo Park, CA, USA. AAAI Press.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto for agile software development.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Cadez, I., Heckerman, D., Meek, C., Smyth, P., and White, S. (2000). Visualization of navigation patterns on a web site using model based clustering. Technical Report MSR-TR-2000-18, Microsoft Research.
- Carroll, J. M. and Rosson, M. B. (1987). Interfacing thought: Cognitive aspects of human-computer interaction. chapter Paradox of the Active User, pages 80–111. MIT Press, Cambridge, MA, USA.
- Chirita, P. A., Firan, C. S., and Nejd, W. (2006). Summarizing local context to personalize global web search. In *Intl. CIKM Conference on Information and Knowledge Management, Arlington, Virginia, USA, 06-11.11.06*.

- Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C. (2005). Using odp metadata to personalize search. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05*, pages 178–185, New York, NY, USA. ACM.
- Chulyadyo, R. and Leray, P. (2014). A personalized recommender system from probabilistic relational model and users' preferences. *Procedia Computer Science*, 35(0):1063 – 1072. Knowledge-Based and Intelligent Information Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.
- Cohen, W. W., Schapire, R. E., and Singer, Y. (1999). Learning to Order Things. *Journal of Artificial Intelligence Research*, 10(1):243–270.
- Cunha, Antonio Gaspar, c., Takahashi, Ricardo, c., and Antunes, Carlos Alberto Henggeler de Carvalho, c. (2012). *Manual de computação evolutiva e metaheurística*. Imprensa da Universidade de Coimbra, Coimbra.
- Dourish, P. (2004). What We Talk About When We Talk About Context. *Personal and Ubiquitous Computing*, 8(1):19–30.
- Eickhoff, C., Collins-Thompson, K., Bennett, P. N., and Dumais, S. (2013). Personalizing atypical web search sessions. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM '13*, pages 285–294, New York, NY, USA. ACM.
- Felfernig, A. and Burke, R. (2008). Constraint-based recommender systems: Technologies and research issues. In *Proceedings of the 10th International Conference on Electronic Commerce, ICEC '08*, pages 3:1–3:10, New York, NY, USA. ACM.
- Freeman, J. A. and Skapura, D. M. (1991). *Neural Networks: Algorithms, Applications, and Programming Techniques*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
- Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press, New York, NY, USA, 1st edition.
- Johnson, D. B. (1973). A note on dijkstra's shortest path algorithm. *J. ACM*, 20(3):385–388.
- Kim, H. R. and Chan, P. K. (2003). Learning implicit user interest hierarchy for context in personalization. In *Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI '03*, pages 101–108, New York, NY, USA. ACM.
- Kimball, R. and Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley Publishing, 3rd edition.
- Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *informatica* 31:249–268.
- Liang, S., Ren, Z., and de Rijke, M. (2014). Personalized search result diversification via structured learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 751–760, New York, NY, USA. ACM.
- Lindenbaum, M. and Samet, H. (1990). Probabilistic analysis of geometric hierarchical data structures. In *Proc. 10th Int. Conf. Pattern Recognition (ICPR)*, Atlantic City.

- Ling, M. (2006). *Web Content Personalization and Task Complexity in e-Commerce Decision Making*. PhD thesis, Lincoln, NB, USA. AAI3237055.
- Liu, F., Yu, C., and Meng, W. (2002). Personalized web search by mapping user queries to categories. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02*, pages 558–565, New York, NY, USA. ACM.
- Liu, J., Dolan, P., and Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI '10*, pages 31–40, New York, NY, USA. ACM.
- Masurel, P. and Lefèvre-Hasegawa, K. (2014). Dataiku’s solution to yandex’s personalized web search challenge. In *Dataiku’s Solution to Yandex’s Personalized Web Search Challenge, WSDM '14*, pages 285–294, New York, NY, USA. ACM.
- McSherry, D. (2005). Explanation in recommender systems. *Artif. Intell. Rev.*, 24(2):179–197.
- Melville, P. and Sindhvani, V. (2010). Recommender systems. In *Encyclopedia of Machine Learning*, pages 829–838.
- Miner, G., Elder, J., Hill, T., Nisbet, R., Delen, D., and Fast, A. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Academic Press, 1st edition.
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.
- Mitchell, T. M. (1997a). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.
- Mitchell, T. M. (1997b). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.
- Papagelis, M., Plexousakis, D., and Kutsuras, T. (2005). Alleviating the Sparsity Problem of Collaborative Filtering using Trust Inferences. In *Proc. of the 3rd International Conference on Trust Management (iTrust'05)*, pages 224–239, Berlin, Heidelberg. Springer-Verlag.
- Pretschner, A. and Gauch, S. (1999). Ontology based personalized search. pages 391–398.
- Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *PROCEEDINGS OF THE IEEE*, pages 257–286.
- rae Kim, H. and Chan, P. K. (2005). Personalized ranking of search results with learned user interest hierarchies from bookmarks. In *In Proc. of WebKDD 2005: KDD Workshop on Web Mining and Web Usage Analysis, in conjunction with the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2005)*, pages 32–43.
- Rasmusson, J. (2010). *The Agile Samurai: How Agile Masters Deliver Great Software*. Pragmatic Bookshelf, 1st edition.

- Rising, L. and Janoff, N. S. (2000). The scrum software development process for small teams. *IEEE Softw.*, 17(4):26–32.
- Robinson, T. (2013). Personalization 101: Types of personalization. <http://http://devnet.kentico.com/articles/personalization-101--types-of-personalization>.
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229.
- Santos, O., Kravcik, M., and Pérez-Marín, D. (2012). Personalization approaches in learning environments. In Ardissono, L. and Kuflik, T., editors, *Advances in User Modeling*, volume 7138 of *Lecture Notes in Computer Science*, pages 117–121. Springer Berlin Heidelberg.
- Schein, A., Popescul, A., Ungar, L., and Pennock, D. (2002). Methods and Metrics for Cold-Start Recommendations. In *Proc. of the 25th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '02)*, pages 253–260, New York, NY, USA. ACM.
- Sheikh, R. H., Raghuwanshi, M., and Jaiswal, A. N. (2008). Genetic algorithm based clustering: A survey. *Emerging Trends in Engineering and Technology, International Conference on*, 0:314–319.
- Shen, X., Tan, B., and Zhai, C. (2005a). Context-sensitive information retrieval using implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA. ACM Press.
- Shen, X., Tan, B., and Zhai, C. (2005b). Implicit user modeling for personalized search. In *In Proceedings of CIKM*.
- Skillen, K.-L., Chen, L., Nugent, C. D., Donnelly, M. P., Burns, W., and Solheim, I. (2012). Ontological user profile modeling for context-aware application personalization. In *Proceedings of the 6th International Conference on Ubiquitous Computing and Ambient Intelligence, UCAmI'12*, pages 261–268, Berlin, Heidelberg. Springer-Verlag.
- Sugiyama, K., Hatano, K., and Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 675–684, New York, NY, USA. ACM Press.
- Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., and Chen, Z. (2005). Cubesvd: A novel approach to personalized web search. In *Proceedings of the 14th International Conference on World Wide Web, WWW '05*, pages 382–390, New York, NY, USA. ACM.
- Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 2–9, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Tan, B., Shen, X., and Zhai, C. (2006). Mining long-term search history to improve search accuracy. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06*, pages 718–723, New York, NY, USA. ACM.

- Teevan, J., Dumais, S. T., and Horvitz, E. (2005a). Beyond the commons: Investigating the value of personalizing web search. In *In Proceedings of the Workshop on New Technologies for Personalized Information Access*, pages 84–92.
- Teevan, J., Dumais, S. T., and Horvitz, E. (2005b). Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05*, pages 449–456, New York, NY, USA. ACM.
- Teevan, J., Liebling, D. J., and Geetha, G. R. (2011). Understanding and predicting personal navigation. In King, I., Nejdl, W., and Li, H., editors, *WSDM*, pages 85–94. ACM.
- Whitley, L. D. (1989). The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 116–123, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 2nd edition.
- Zimmermann, A., Lorenz, A., and Oppermann, R. (2007). An Operational Definition of Context. In *Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2007)*, volume 4635 of *Lecture Notes in Computer Science*, pages 558–571, Roskilde, Denmark. Springer.