FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Data Mining Teams: Plataforma Colaborativa para Projetos de Data Mining

Ana Sara Videira Morais

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Professor Doutor Carlos Soares

Co-orientador: Professor Doutor Rui Rodrigues

11 de julho de 2014

Data Mining Teams: Plataforma Colaborativa para Projetos de Data Mining

Ana Sara Videira Morais

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Prof. Henrique Daniel de Avelar Lopes Cardoso

Arguente: Prof. Célia Talma Martins de Pinho Valente Oliveira Gonçalves

Vogal: Prof. Carlos Manuel Milheiro de Oliveira Pinto Soares

---

11 de julho de 2014

Resumo

Este projeto centra-se na problemática do trabalho em equipa na área de *Data Mining*. Em particular, esta dissertação tem como principal preocupação a utilização de ferramentas diferentes de análise de dados, por diferentes elementos da equipa e na dificuldade da gestão da informação gerada nessas análises, por causa das diferenças nos formatos respetivos. Assim, no sentido de contribuir para a resolução deste problema foi proposto especificar, desenvolver e implementar o protótipo, de uma plataforma colaborativa que auxilie a partilha de dados em projetos de Data Mining.

Acerca do desenvolvimento da solução, a plataforma Data Mining Teams (DMT), salientam-se a metodologia de desenvolvimento assente nas etapas: Conceção (levantamento de requisitos e arquitetura), Desenvolvimento (desenho da interface e implementação) e os Testes (testes e *deployment*). O levantamento de requisitos teve como suporte a realização de entrevistas a especialistas em DM e questionários a pessoas que trabalham na área do DM. Relativamente à arquitetura especificam-se os aspetos associados à arquitetura física, arquitetura lógica e arquitetura tecnológica, bem como estilos e padrões de arquitetura. Foi possível obter *feedback* do protótipo funcional da plataforma com uma pequena experiência de resolução de um problema de uma competição de *Data Mining*, realizado por duas equipas, uma fazendo uso da DMT e outra utilizando apenas as ferramentas tradicionais.

As experiências realizadas, embora limitadas, mostram que o projeto aborda um problema importante e que a abordagem seguida é adequada, levando a que o trabalho em equipa, em projetos de Data Mining, seja mais fluído. Permitiram também identificar alguns problemas e limitações do protótipo desenvolvido bem como novas funcionalidades que será desejável implementar.

Abstract

This project focuses on the issue of teamwork in the Data Mining (DM) area. In particular, this dissertation's main concern is the use of different tools for data analysis, for different members of the team and the difficulty of managing information generated in these analyzes, because of differences in their respective formats. Thus, in order to contribute to solving this problem, it has been proposed to develop and implement a prototype of a collaborative platform that helps data sharing in data mining projects.

About the development of the Data Mining Teams (DMT) platform as a solution to this problem, we emphasize the methodology of development based on stages: Conception (requirements gathering and architecture), development (interface design and implementation) and tests (testing and deployment). The requirements gathering was supported by interviews with experts in DM and questionnaires to people working in the DM area. Regarding the architecture, we specifed the aspects associated with physical architecture, logical architecture and technology architecture as well as architectural styles and patterns. It was possible to obtain feedback from the working prototype platform with a small-scale experiment of solving a problem in a data mining competition held by two teams, one making use of DMT and another using only the traditional tools.

The experiments, although limited, show that the project addresses a major problem and that the approach is appropriate, leading to team work in data mining projects being more fluid. It also allowed to identify some problems and limitations of the developed prototype and new features that will be desirable to implement.

Agradecimentos

Após esta longa e trabalhosa jornada não posso deixar de agradecer aqueles que estiveram presentes e tanto me apoiaram:

Aos meus orientadores Professor Carlos Soares e Professor Rui Rodrigues pela disponibilidade, pelo acompanhamento e pelas críticas e sugestão, em particular ao Professor Carlos pela paixão que sempre demonstrou pelo projeto que me levou a acreditar que era realmente possível de realizar, e ao Professor Rui pelos conselhos práticos que me ajudaram a traçar uma solução;

Aos membros da DIG, em particular, ao Pedro Abreu, à Catarina Félix e ao Fábio Pinto pela disponibilidade e colaboração nos testes à plataforma;

Ao Carlos Carvalheira pelo seu tempo e pela ajuda que me deu;

Ao Luís Fonseca e ao Ricardo Amorim pelo companheirismo, as brincadeiras, a amizade, a ajuda e a força durante estes últimos meses;

À Maria pela sua amizade, ajuda e força incondicional;

À minha irmã pela paciência de aturar a minha falta de paciência, pelo carinho e amizade;

Aos meus pais por acreditarem em mim, pela dedicação, os conselhos sábios e o carinho que sempre me deram;

Por fim, a todos os meus amigos, que durante estes últimos anos me fizeram crescer, estiveram presentes nos bons e nos menos bons momentos e nunca me deixaram desistir.

Um muito obrigada a todos.

Sara Morais

... aos meus Pais

"If you can dream it, you can do it."
Walt Disney

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

CRAN	Comprehensive R Archive Network
CRISP-DM	Cross-Industry Standard Process for Data Mining
CRUD	Create, Read, Update, Delete
DM	Data Mining
DMT	Data Mining Teams
FEUP	Faculdade de Engenharia da Universidade do Porto
GNU	GNU's Not Unix
HTML	HyperText Markup Language
IDE	Integrated Development Environment
INESC TEC	Instituto de Engenharia de Sistemas e Computadores Tecnologia e Ciência
ISO	International Standards Organization
KDD	Knowledge Discovery in Database
LIACC	Laboratório de Inteligência Artificial e Ciência dos Computadores
MVC	Model-View-Controller
RUP	Rational Unified Process
SEMMA	Sample, Explore, Modify, Model, Assess
SQL	Structured Query Language
UML	Unified Modeling Language
URL	Uniform Resource Locator
XML	eXtensible Markup Language

Capítulo 1

Introdução

Nos dias de hoje, as instituições confrontam-se, diariamente, com uma grande quantidade de dados, provenientes de várias fontes, dos quais têm necessidade de extrair informação e conhecimento úteis. A informação e o conhecimento extraídos são necessários para apoiar a tomada de decisões. A análise de dados e extração de informação para construção de conhecimento são tarefas realizadas, na maioria das vezes, por equipas de *Data Mining*, muitas vezes multidisciplinares. Estas tarefas exigem a colaboração de vários intervenientes, nomeadamente na troca de dados, dos modelos a adotar e dos resultados de avaliação. No entanto, as ferramentas de DM existentes ainda necessitam de ser melhoradas para facilitar essa colaboração.

Há diversas ferramentas que podem ser usadas para fazer a análise de dados, cada uma com vantagens relativamente a alguns aspetos e desvantagens relativamente a outros. Dentro de cada equipa de trabalho são, frequentemente, usadas várias ferramentas diferentes que geram resultados em diferentes formatos. Os dados e modelos em formatos diferentes dificultam a colaboração e condicionam a eficiência do processo de partilha entre os membros das equipas. Este problema leva a que uma parte significativa do tempo seja gasta no processo de conversão entre dados/modelos de diferentes ferramentas.

Assim, este projeto tem como principal objetivo investigar, especificar e desenvolver uma plataforma que agregue a informação gerada por vários utilizadores, após a análise de dados no processo de DM, com uma interface única, interativa e de fácil utilização. Segue-se o desenvolvimento dos tópicos mais relevantes para a implementação do projeto e da dissertação, nomeadamente, o problema e enquadramento, objetivos e estrutura da dissertação.

1.1 Problema e Enquadramento do Projeto

Tem havido um crescimento notório de competições associadas à resolução de problemas reais, nas áreas de *Analytics, Data Science e Data Mining* (DM). Este projeto surgiu de uma dificuldade real, sentida por uma equipa constituída por profissionais de DM da FEUP[1], INESC TEC[2],

[1] http://www.fe.up.pt
[2] http://www.liacc.up.pt/

LIACC[3] e Labs Sapo[4], durante a participação numa dessas competições. Dentro desta equipa existem sub-grupos de trabalho que utilizam diferentes ferramentas para a análise de dados. Após a análise dos dados, os resultados obtidos são apresentados ao resto da equipa recorrendo a reuniões presenciais, o que torna o processo muito mais moroso. A solução deste tipo de problemas exige um trabalho recursivo de manipulação dos dados, sendo que a cada iteração da análise feita esses resultados são discutidos e apresentados aos restantes membros da equipa. Todo este processo pode levar a atrasos e dificuldades na partilha dos dados. Em conversa com peritos da área de DM e posteriormente confirmado pelas entrevistas realizadas para o levantamento de requisitos foi possível observar que é um problema recorrente em projetos de DM, não sendo um problema específico da participação em competições. Tendo em vista a contribuição para a resolução deste problema, foi proposto o desenvolvimento deste projeto, Data Mining Teams (DMT): Plataforma Colaborativa para Projetos de Data Mining.

1.2 Objetivos do Projeto

Os principais objetivos deste projeto são desenhar e construir uma plataforma, com um ambiente gráfico interativo e colaborativo, através da qual seja possível:

- Importar dados e resultados, independentemente da ferramenta em que foram gerados;
- Exportar dados e resultados que fiquem imediatamente disponíveis para ferramentas, independente dos formatos que utilizem;
- Documentar transformações e análises de dados ou conjuntos de dados.

Para atingir os objetivos referidos, propõe-se que a plataforma faça a integração das diferentes ferramentas de DM. A plataforma deve facilitar a utilização e reutilização de dados e resultados por diferentes elementos das equipas de utilizadores e em diferentes contextos.

Pretende-se ainda, após a concretização dos objetivos referidos, proceder à avaliação da plataforma numa equipa de projetos de DM. O objetivo é testar a usabilidade da sua interface gráfica e o quão intuitiva é a manipulação dos dados para utilizadores.

1.3 Estrutura da Dissertação

A dissertação está organizada em seis capítulos, designados por: Introdução, Estado da Arte, Especificação e Arquitetura da Plataforma DMT, Interface e Implementação da Plataforma DMT, Testes e Resultados e Conclusões e Trabalho Futuro. Termina com as Referências Bibliográficas e Anexos.

O capítulo 2, Estado da Arte, é dedicado à revisão bibliográfica e está dividido em duas secções principais: *Data Mining* (DM) e Interação Pessoa-Computador. A secção *Data Mining* começa

[3] http://labs.sapo.pt/
[4] http://www2.inescporto.pt/

com uma introdução aos conceitos básicos de *Data Mining* e salientam-se as principais metodologias desta área, prosseguindo com a exposição das características de algumas ferramentas de *Data Mining*. Ainda nesta secção, é apresentado um levantamento das ferramentas mais atuais de desenvolvimento colaborativo de projetos, e apresentadas algumas das suas características e potencialidades, terminando com a análise dos domínios de aplicação das mesmas e dos benefícios do uso destas plataformas. A última parte deste capítulo, *Interação Pessoa-Computador*, inicia-se com a definição de alguns conceitos, apresenta-se a importância desta interação, continuando com a descrição da metodologia e da apresentação das características a ter em conta relativamente aos perfis de utilizadores finais de uma plataforma.

No capítulo 3, Especificação e Arquitetura da Plataforma DMT, apresentam-se as fases de levantamento e especificação de requisitos, bem como a arquitetura da DMT. No levantamento de requisitos sobressaem os procedimentos efetuados, salientando-se a recolha de dados por inquérito a especialistas e a pessoas familiarizadas com trabalhos na área de DM. Na secção dedicada à arquitetura da plataforma são especificados os padrões de arquitetura presentes na DMT, apresentado o desenho da arquitetura e são detalhadas as arquiteturas física, lógica e tecnológica.

No capítulo 4, Interface e Implementação da Plataforma DMT, são apresentados os *mockups* e descritas as principais decisões tomadas relativamente ao desenho da interface da plataforma. Ainda neste capítulo, é discriminado o ambiente de desenvolvimento e apresentado o protótipo funcional, ou seja, a implementação da plataforma, evidenciando os aspetos concretizados com sucesso, bem como aqueles que necessitam de ser repensados e melhorados.

No capítulo 5, Testes e Resultados, apresenta-se a metodologia usada para testar o protótipo funcional desenvolvido da plataforma, assim como os resultados dessa avaliação.

Por fim, no capítulo 6, Conclusões e Trabalho Futuro, é feita uma síntese da concretização dos objetivos do projeto, são expostas as mais valias do trabalho e apresentam-se as possíveis melhorias a efetuar no trabalho realizado.

Capítulo 2

Estado da Arte

Neste capítulo é apresentada a bibliografia analisada sobre os principais temas envolvidos no desenvolvimento deste projeto. Está dividido em duas secções, correspondentes às áreas estudadas: *Data Mining* (DM) e Plataformas de Desenvolvimento Colaborativas de Projetos, e Interação Pessoa-Computador.

Sendo este projeto o estudo e desenvolvimento de uma plataforma para projetos de DM é importante o domínio do conceito e das etapas do processo de DM (metologias de DM) que justificam esta necessidade. Com a pretensão de que seja uma plataforma colaborativa, foram estudadas e é apresentada uma pequena síntese com algumas das plataformas colaborativas existentes no mercado, tendo em conta as limitações na integração de ferramentas de DM, justificando assim a impossibilidade de individualmente solucionarem o problema proposto neste projeto. Por fim, a exigência de que tenha uma interface gráfica com elevada usabilidade, cria a necessidade de aprofundar o conhecimento na área de interação pessoa-computador.

2.1 Data Mining

Data Mining é a área da Ciência da Computação que, descrita de uma forma genérica e simplificada, conduz à tomada de decisões tendo por base a análise de dados históricos. O conceito de DM é complexo envolvendo muitas e diversificadas dimensões, dependendo da área de atuação. Nesta secção são mostradas várias definições e perspetivas sobre DM, com base nos estudos de vários autores. É apresentada com mais detalhe a metodologia CRISP-DM e ainda ferramentas utilizadas para análise dos dados nesta área.

Witten [WF05], Frank [WF05], Olson [OD08], Deten [OD08] e Bramer [Bra07] evidenciam algumas das áreas nas quais a DM é aplicada e apresenta resultados satisfatórios:

- Retenção de clientes e *Telemarketing*: identificação de perfis de clientes, que permite a formação de segmentos, para apoiar na decisão de campanhas publicitárias sobre quais os produtos mais adequados a cada segmento;

- Bancos: identificação de padrões de comportamentos ou desvio destes, para auxiliar na deteção de fraudes;
- Informação Eleitoral: identificação de perfis para possíveis votantes e resultados de campanhas eleitorais antigas para prever resultados de eleições futuras;
- Medicina: auxílio na identificação de diagnósticos mais precisos.

2.1.1 Conceitos e Definições

Assiste-se nos dias de hoje a um aumento desmesurado de geração de dados e armazenamento dos mesmos, incremento que tem vindo desde há quase duas décadas. Goebel e Gruendwald [GG99] concluíram que é muito complexo e difícil o processamento dos dados através dos métodos tradicionais. Aparece, assim, a necessidade da criação de novas técnicas e ferramentas computacionais que auxiliem na extração de informação útil, ou seja, informação que possa ser transformada em conhecimento. Deste modo, há um aprofundamento do conceito de *Knowledge Discovery in Database* (KDD), que recorre a modelos e técnicas de DM para a extração da informação útil, padrões e tendências de forma autónoma e semi-automática.

No sentido de facilitar a compreensão e aprofundamento do conceito de DM, apresentam-se algumas opiniões identificadas na literatura sobre o tema.

Segundo Han e Kamber [HK06], DM é uma dos passos integrantes do processo de KDD que consiste numa sequência iterativa de passos (*Cleaning and Integration, Selection and Transformation, Data Mining e Evaluation and Presentation*). Esta ideia é bastante próxima da de Hand [Han07] que considera DM como a análise de grandes quantidades de dados com a finalidade de encontrar relações inesperadas e de resumir os dados de forma a que eles sejam ao mesmo tempo úteis e compreensíveis aos seus utilizadores. Com uma visão mais ligada especificamente às bases de dados, Cabena et al [CHS+98] referem que DM é o campo interdisciplinar que junta técnicas de reconhecimento de padrões, estatística, bases de dados e visualização para conseguir extrair informações de grandes bases de dados.

Larose [Lar04] define seis tipos de tarefas que podem ser abordadas no processo de DM: descrição, classificação, estimação, previsão, *clustering* e associação.

- Descrição é a tarefa usada, muitas vezes, juntamente com técnicas de análise exploratória de dados, para verificar os resultados obtidos.
- Classificação é a tarefa que tem como objetivo prever qual a classe a que um determinado registo pertence. Sendo uma das tarefas mais comuns em DM, a classificação é a tarefa cujo modelo é obtido da análise de um conjunto de registos fornecidos, com cada registo já analisado e colocado na classe pertencente.
- Estimação é uma variante do problema de Classificação, na qual é atribuída uma pontuação a cada registo. Desta forma é possível estimar o valor de uma determinada variável analisando o valor das restantes.

- Previsão é uma tarefa semelhante às tarefas de Estimação e Classificação, que tem como fim antever o valor futuro de um determinado atributo.

- *Clustering* é a tarefa do processo de DM que tem como finalidade a identificação e agrupamento dos registos de dados, observações ou casos em classes de objetos similares. Um *cluster* é um agregado de registos de dados semelhantes entre si, mas diferentes dos registos dos restantes agrupamentos. A tarefa de *Clustering* não exige a categorização prévia dos registos, nem pretende classificar, estimar ou predizer o valor de uma variável. *Clustering* apenas identifica os conjuntos de dados semelhantes.

- Associação é a tarefa de DM que pretende encontrar relações entre valores de diferentes variáveis. Um exemplo conhecido é *Market Basket Analysis* em que cada registo representa um cesto de compras e o objetivo é encontrar produtos que têm tendência a serem comprados em conjunto. Assim, o objetivo é descobrir tendências nas várias transações estudadas de modo a entender o comportamento dos clientes e encontrar padrões de compras.

2.1.2 Metodologias de Data Mining

O processo de DM é mais fácil de compreender, implementar e desenvolver quando enquadrado numa metodologia. Para tal, foram desenvolvidas várias metodologias, por vários autores, das quais se podem destacar: a metodologia *CRoss Industry Standard Process for Data Mining* (CRISP-DM), a metodologia *Sample, Explore, Modify, Model and Assess* (SEMMA), criada pela SAS [1], a metodologia de Pechenizkiy [PPT08] que propõe um processo baseado num modelo de Sistemas de Informação e a metodologia proposta por González [GARM+08] que apresenta um modelo que assenta no processo de *Rational Unified Process* (RUP).

Segundo Costa [Cos11], baseados em Mainon e Rokach [MR10], a metodologia CRISP-DM surgiu na tentativa de padronizar o processo de DM, tendo sido construída com base no conhecimento académico dos seus autores e também na experiência e conhecimentos que adquiriram ao longo dos anos. Também no estudo realizado por Costa [Cos11] a metodologia CRISP-DM apresenta vantagens quando comparada com a metodologia SEMMA, mostrando-se como uma metodologia neutra, no que diz respeito à adoção de ferramentas de DM. Por estas razões, e pelo facto de ser uma das mais populares, este estudo focar-se-á na metodologia CRISP-DM. Esta metodologia foi desenvolvida por um conjunto de empresas, entre as quais SPSS [2].

A metodologia CRISP-DM é descrita como um modelo de processos hierarquizados: *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, *Deployment*. Trata-se de uma metodologia com várias fases, organizadas ciclicamente, com um fluxo não unidirecional, como pode ser observado na Figura 2.1.

As seis fases que constituem o Modelo CRISP-DM, de acordo com Chapman et al [CCK+00], são as seguintes:

[1] http://www.sas.com
[2] http://www-01.ibm.com/software/analytics/spss/

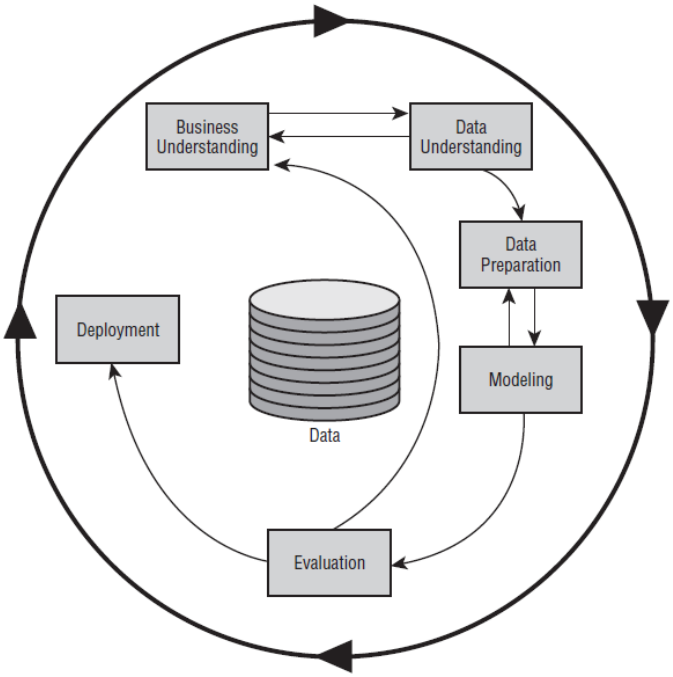


Figura 2.1: Fases do Modelo CRISP-DM (extraído de CRISP-DM 1.0, 2000, p.10)

- ***Business Understanding***: fase com foco na compreensão dos objetivos e requisitos do projeto, numa perspetiva de negócio, para posteriormente converter esse conhecimento num problema de DM. Esta é uma fase essencial para uma melhor compreensão das próximas.

- ***Data Understanding***: a interpretação dos dados começa com a recolha destes, seguindo-se uma análise detalhada que permite ao utilizador uma maior familiarização com os mesmos.

- ***Data Preparation***: esta fase envolve a transformação, "limpeza"de tabelas, registos e atributos dos dados que leva à construção final dos conjuntos de dados, que será utilizado por ferramentas de modelação.

- ***Modeling***: fase em que são selecionadas e aplicadas as técnicas e algoritmos de DM, sendo os seus parâmetros calibrados de forma a atingir o valores otimizados.

- ***Evaluation***: fase em que os modelos obtidos são avaliados face ao contexto em que vão ser usados e aos objetivos propostos inicialmente.

- ***Deployment***: esta é a última fase do processo, dedicada à organização do conhecimento extraído dos modelos, para ser posteriormente apresentado aos clientes ou integrado nos sistemas para apoio à decisão. Esta fase destina-se ainda à documentação do projeto.

Mendes [Men11] afirma que “a CRISP-DM não garante resultados mas permite disciplinar o processo e tem como grande finalidade alinhar os objetivos de DM com o negócio".

2.1.3 Ferramentas de Data Mining

O mercado oferece um vasto conjunto de ferramentas, sendo muitas delas gratuitas, de *open-source*, das quais serão apresentadas R[3], Weka[4], RapidMiner[5] e KNIME[6]. Algumas destas ferramentas têm preocupações quanto à facilidade de uso das mesmas, tentando tornar a aplicação de DM uma tarefa menos técnica, ou seja, uma tarefa acessível a profissionais de várias áreas.

Deve ter-se presente que a escolha de uma ferramenta de DM é uma tarefa complexa. Costa [Cos11] alerta para a importância de ter em conta diversos fatores, como por exemplo se é para fins académicos ou comerciais, o domínio do problema, sistema operativo, custo, licença e o tipo de uso.

Segundo Goebel e Gruendwald [GG99] as características determinantes a ter em conta sobre o desempenho das ferramentas de DM são: capacidade de aceder às fontes de dados, acesso *online/offline* aos dados, modelos de dados, número de tabelas, linhas e atributos, tamanho da base de dados com que a ferramenta de DM é compatível, consultas e tipos de atributos com que a ferramenta pode lidar.

Com base em questionários *online* disponibilizado no *site kdnuggets.com*, Fayyad, Piatetsky-Shapiro e Smyth realizaram uma análise sobre as ferramentas de DM mais usadas pela comunidade de DM. O estudo mostra que as ferramentas de DM *open-source* estão no topo das preferências, destacando-se: RapidMiner, o R e Knime, como se pode ver na Figura 2.2.

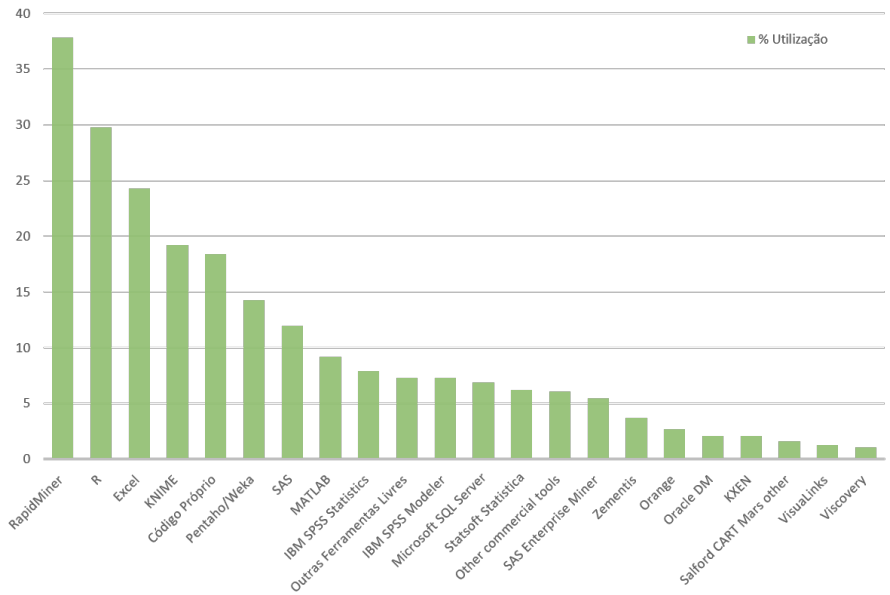


Figura 2.2: Ferramentas mais usadas na comunidade DM. Adaptado de [Piatetsky-Shapiro, 2010]

[3] http://http://www.rdatamining.com/
[4] http://www.cd.waikato.ac.nz/ml/weka/
[5] http://www.rapidminer.com/
[6] http://www.knime.org/

Segue-se um breve sumário de algumas das ferramentas mais usadas, referidas anteriormente.

R

O R é um ambiente de programação criado essencialmente para o desenvolvimento de sistemas de apoio à decisão e análise de dados. Trata-se de um projeto GNU e foi desenvolvido na Universidade de Auckland, na Nova Zelândia. Está preparado para ser instalado em qualquer plataforma (*Windows, Mac OS, Linux*) e disponibiliza uma vasta quantidade de *packages*.

O R é um conjunto integrado de software que permite ao utilizador a manipulação de dados, cálculo e representação gráfica, através de um ambiente de programação de linha de comandos. São desenvolvidas novas bibliotecas, frequentemente, para resolução de novos problemas. O repositório CRAN contém um conjunto de bibliotecas que possibilita aos seus utilizadores integrarem no ambiente R aplicações escritas noutras linguagens de programação, o que é o caso de Java, C, C++ e Perl, o que leva a um aumento de potencialidades. Esta ferramenta suporta vários aspetos de DM, nomeadamente muitos algoritmos incluindo árvores de regressão e classificação (biblioteca rpart), modelos de regressão linear (função lm) e redes neuronais (biblioteca nnet).

Na Figura 2.3 é possível ver um exemplo do ambiente R.

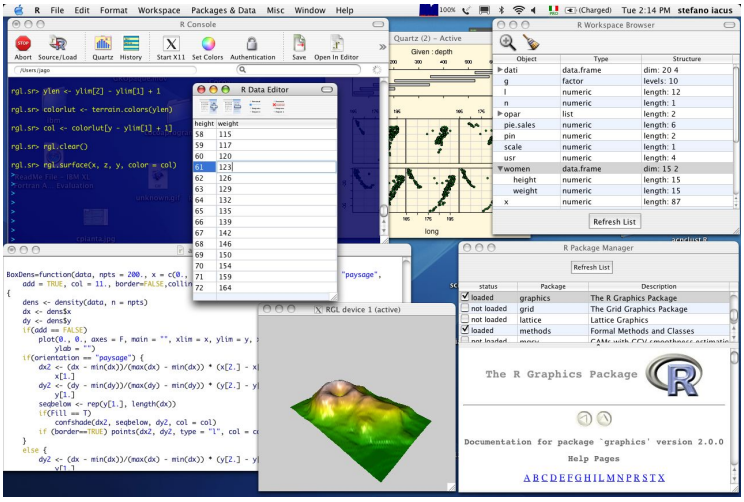


Figura 2.3: Exemplo do ambiente de programação R (recuperado de http://www.r-project.org/screenshots/)

Weka

Criada no ano de 1993, a ferramenta Weka (*Environment for Knowledge Analysis*) é fruto de um financiamento do governo da Nova Zelândia. A aplicação foi desenvolvida na linguagem Java com o objetivo de ser aplicada à economia do país. Atualmente, a Weka é usada tanto por empresas como por universidades. Esta ferramenta disponibiliza ao utilizador as funcionalidades para pré-processamento, classificação, regressão, *clustering*, regras de associação e visualização.

A Weka permite que o utilizador trabalhe diretamente na linha de comandos, disponibilizando-lhe também uma interface gráfica. Esta interface permite ao utilizador ver o *workflow* dos projetos. Na Figura 2.4 apresenta-se um exemplo da aplicação Weka 3.5.5.

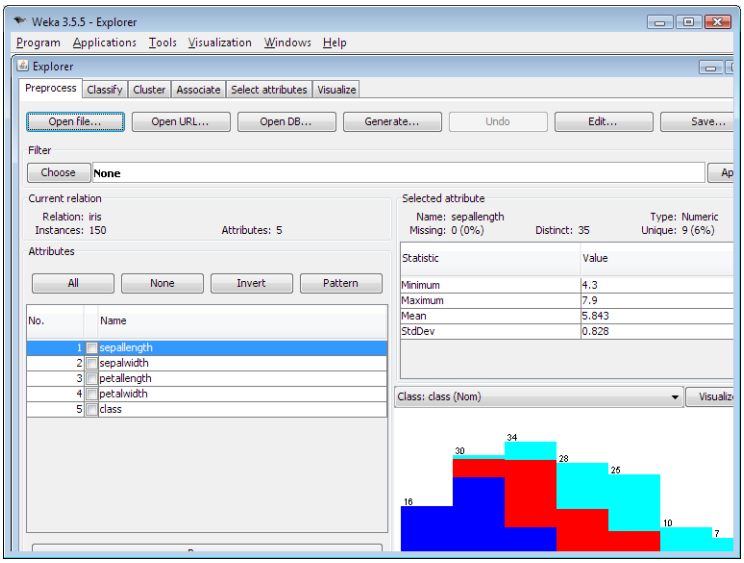


Figura 2.4: Exemplo da aplicação Weka-3.5.5 (recuperado de http://pt.wikipedia.org/wiki/Weka)

RapidMiner

O RapidMiner é uma ferramenta *open-source* para DM das mais utilizadas mundialmente. Costa [Cos11] mostra o conjunto das principais características do RapidMiner:

- Integração de dados, ETL (*Extract, Transform, Load*), análise de informação e produção de relatórios, tudo numa única ferramenta;
- Reconhecimento de erros *on-the-fly* e correções rápidas;
- Escrita em Java;
- As representações internas em XML, asseguram um formato estandardizado dos resultados de DM;

O RapidMiner é uma ferramenta muito usada em DM pelo elevado número de modelos que suporta, por disponibilizar mais de 1500 operadores/funções para o tratamento do dados e pela sua interface para o utilizador de elevada usabilidade. Na Figura 2.5 é possível ver um exemplo da aplicação RapidMiner.

KNIME

KNIME (*Konstanz Information Miner*) é também uma ferramenta *open-source* que oferece ao utilizador funcionalidades que permitem a integração, processamento e análise de dados. Na Figura 2.6 é possível ver um exemplo da aplicação Knime.

A Tabela 2.1 é uma síntese da informação mais relevante referentes às ferramentas apresentadas. Foi possível concluir que todas as ferramentas permitem a análise e transformação de dados utilizando vários métodos, as não permitem colaboração.

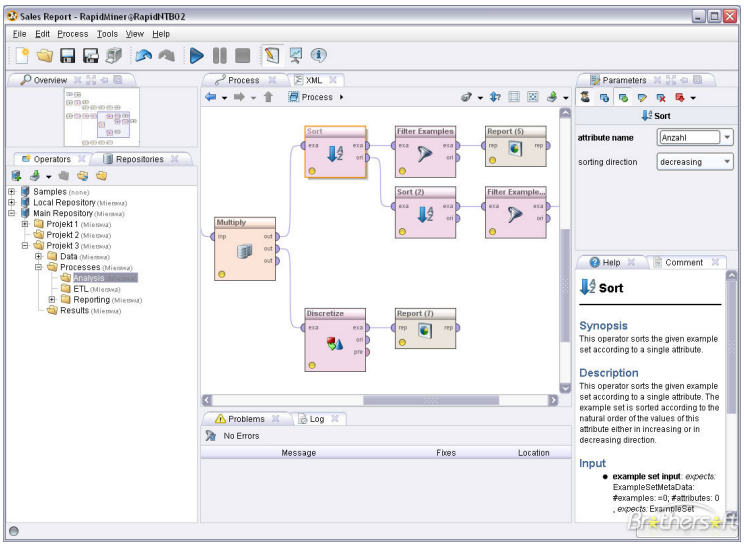


Figura 2.5: Exemplo da ferramenta RapidMiner (recuperado de http://www.brothersoft.com/rapidminer-165969.html)

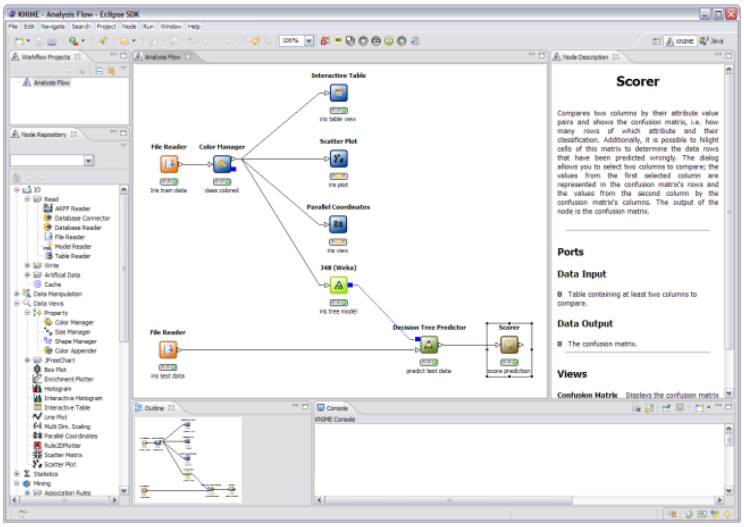


Figura 2.6: Exemplo da ferramenta KNIME (recuperado de http://www.knime.org/node/919)

2.1.4 Ferramentas de Desenvolvimento Colaborativo de Projetos

As ferramentas colaborativas de desenvolvimento de projetos são um tipo específico de ferramentas, para ajudar em projetos de Engenharia de Software, que abrangem áreas como: *Bug Tracking*, Desenvolvimento, *Design*, Gestão e Engenharia de Requisitos. Assim, há no mercado um vasto leque de ferramentas *open-source* que auxiliam a gestão dos projetos, nas suas diferentes componentes: tarefas, distribuição de trabalho, gestão de colaboradores, gestão do desenvolvimento (código, qualidade de código, arquitetura) e ainda gestão de testes.

Ferramenta	Linguagem de Programação	Interface Gráfica Linha de Comandos	Principal Propósito
R	Java, C, C++, Perl, R, Fortran	Ambos	Sistemas de apoio à decisão e Estatística
Weka	Java	Ambos	Data Mining
RapidMiner	Java	Interface Gráfica	Data Mining
Knime	Java	Interface Gráfica	Data Mining

Tabela 2.1: Principais Características das Ferramentas R, Weka, RapidMiner e Knime [Sah]

Teixeira [Tei09] apresenta no seu estudo a comparação entre várias ferramentas de colaboração, de acordo com três critérios: nível de colaboração, no que diz respeito a saber o ponto de situação do projeto, comunicação entre os intervenientes e passagem de conhecimento; integração, relativamente à possibilidade de integração de outras ferramentas e *plugins*; e outras características, nomeadamente recursos utilizados. Teixeira [Tei09] concluiu que a maioria das ferramentas de colaboração são ferramentas *web*, enquanto que muitas das ferramentas usadas, por exemplo, para desenho de diagramas UML são ferramentas de *desktop* sem funcionalidades de colaboração. Este indica como características com potencial para apoiar atividades colaborativas: os comentários, gestão de contactos, partilha de ficheiros, partilha de diretórios, *online whiteboard*, *chat*, sistema de recomendações, partilha de ecrã, *timeline* das atividades da equipa, lista de tarefas, sistema de votação para classificação das melhores e piores funcionalidades da ferramenta e controlo de versões.

É possível ver na tabela 2.2 uma comparação de algumas das ferramentas colaborativas mais usadas no mercado, considerando os critérios estabelecidos por Teixeira [Tei09], que se mostram mais relevantes para este projeto.

	Comentários	Partilha de Ficheiros	Timeline de Atividades	Lista de Tarefas	*Chat*	Controlo de Versões
Jira	✓	✗	✗	✗	✗	✓
Eclipse	✗	✓	✓	✗	✓	✓
NetBeans	✓	✗	✗	✗	✓	✓
Microsoft Project	✓	✗	✓	✗	✗	✗
Trac Project	✓	✓	✓	✓	✗	✓
Redmine	✓	✓	✓	✓	✗	✓

Tabela 2.2: Tabela de comparação entre algumas ferramentas colaborativas de desenvolvimento de projetos

Depois da análise feita, pode concluir-se que estas ferramentas não permitem solucionar o problema inerente a este projeto, uma vez que não se adequam às necessidades do processo de DM, visto que não permitem a análise de dados.

2.2 Interação Pessoa-Computador

A Interação Pessoa-Computador é uma disciplina complexa que relaciona diversas áreas como a ciência da computação, artes, design, psicologia, ergonomia entre outras. Esta secção destina-se à apresentação da análise de conceitos, definições, objetivos e algumas metodologias na área da Interação Pessoa-Computador.

2.2.1 Conceitos, Definições e Objetivos

Quando se pensa no desenvolvimento de uma nova plataforma, especificamente no desenho das suas interfaces e interação com o utilizador, é essencial ter presente alguns conceitos fundamentais, nomeadamente: *Design* de Interação, *Design* Centrado no Utilizador e Usabilidade.

O *Design* de interação tem sido um tema alvo de estudo por vários autores como Preece [RSP02], Cooper, Reimann e Cronin [CRC07] e é entendido como o *design* que promove o equilíbrio entre funcionalidades e a simplicidade e facilidade de utilização de uma interface. Deste modo, tem como objetivo, facilitar a interação do utilizador, tendo em conta o contexto para o qual a plataforma está a ser desenvolvida.

De encontro ao conceito anterior também é possível encontrar na literatura o conceito de *Design* Centrado no Utilizador. Com definições muito semelhantes, Norman [Nor04], Alan Dix et al. [DFAB97] e Shneiderman [Shn97] referem que este tipo de *design* tem como foco principal o utilizador do sistema, sendo por isso a criação do mesmo um processo que requer acompanhamento de perto por parte do utilizador, para que facilmente seja ouvido o seu *feedback*. Esta vertente de *design* deve conduzir ao desenvolvimento de sistemas de fácil utilização, mais intuitivos, com melhor desempenho e menos falhas. É um tipo de *design* que visa a diminuição do tempo e do custo de desenvolvimento do sistema, bem como da sua manutenção. Portela [Por12] reforça seis princípios, descritos pela norma ISO 9241-210, que caraterizam o desenvolvimento de *design* centrado no utilizador:

- O *design* de sistemas interativos é baseado num entendimento dos utilizadores, tarefas e ambientes;
- Os utilizadores estão envolvidos em todo o processo de desenvolvimento do *design*;
- *O* design é dirigido e refinado considerando a avaliação dos utilizadores;
- O processo de desenvolvimento é iterativo;
- O *design* especifica toda a experiência do utilizador;
- A equipa de desenvolvimento deve ser multidisciplinar.

Machado [Mac13] apresenta o conceito de usabilidade, na visão de vários autores, como Nielsen [Nie12], Ribeiro [Rib13], Shneiderman [Shn97] entre outros, descritas de seguida. Segundo Nielsen [Nie12], usabilidade é um atributo de qualidade que se usa para avaliar a facilidade que

um utilizador tem na utilização de uma interface, servindo, também, como método para melhorar a facilidade de utilização durante o processo de desenvolvimento do *design*. Por sua vez, Ribeiro [Rib13] define usabilidade como uma forma de medir a capacidade de eficiência com que um sistema satisfaz as necessidades do utilizador, o objetivo para que foi criada e a eficiência de interação para o utilizador.

2.2.2 Metodologias

Para o desenvolvimento de uma plataforma que garanta qualidade relativamente à sua usabilidade e ao seu *design* de interação, foram alvo de estudo algumas metodologias. São apresentadas aquelas que se julga serem as mais adequadas para auxiliarem o desenvolvimento da plataforma que se pretende criar neste projeto.

Caddick e Cable [CC11] definem o conceito de *persona* como:

> "A persona is a document that describes the ways in which certain types of people will use your website. Usually one persona is created for each type of user. "

Tal como a definição refere, o uso de *personas* leva a que o desenvolvimento do *design* da interface se centre no utilizador final do produto. Machado [Mac13] afirma também que permite a criação de modelos tendo em conta, tanto as necessidades do utilizador, como a localização, estado de interesse e tarefas que este pretende ver resolvidas. Envolve um processo complexo e de várias fases desde entrevistas ao público-alvo, à definição de variáveis e criação de modelos.

Machado [Mac13] acrescenta ainda que a criação de *Personas* só fica completa quando associada a objetivos, podendo estes, segundo Cooper [CRC07], ser de quatro tipos, consoante a sua origem: pessoais, corporativos práticos ou falsos. De acordo com Cooper [CRC07]:

> "Personas provide us with a precise way of thinking and communicating about how users behave, how they think, what they wish to accomplish, and why. "

Card-sorting é outra metodologia que complementa a anterior. Segundo Ribeiro [Rib13], possibilita a compreensão da forma como os utilizadores agrupam informação e quais as interpretações que fazem de termos e conceitos. Para tal, como refere Machado [Mac13], é usado um sistema de cartões individuais com um conjunto de terminologias. Existem dois tipos de *card-sorting*: aberto ou fechado. Por exemplo, quando este método é usado para o desenvolvimento de uma página *web* é entregue ao utilizador um conjunto de cartões com o nome de componentes que irão constituir a página, como mostra a figura 2.7.

No *card-sorting* aberto o utilizador tem a liberdade e é responsável por agrupar as secções e classifica-las como achar mais conveniente, sendo ele que dá nome aos *menus*, como é exemplificado na figura 2.8. Quando se trata de *card-sorting* fechado os participantes da experiência são convidados o conteúdo em grupos pré-definidos, dando-lhes para além dos cartões os nomes dos *menus* também.

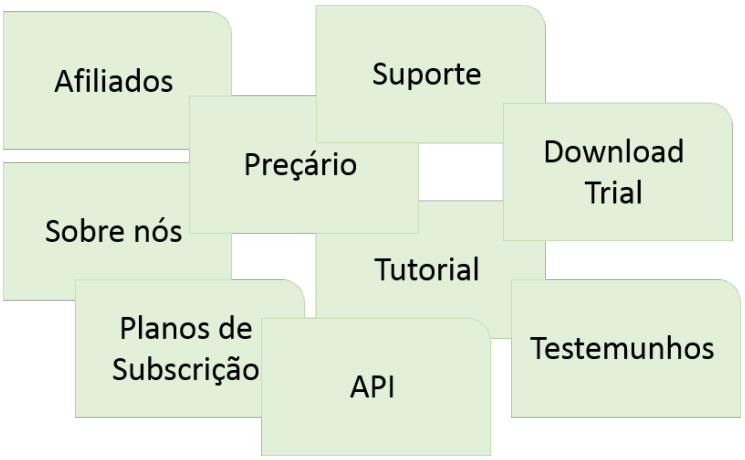


Figura 2.7: Exemplo de Cartões de *Card-Sorting* [Kni, adaptado de]

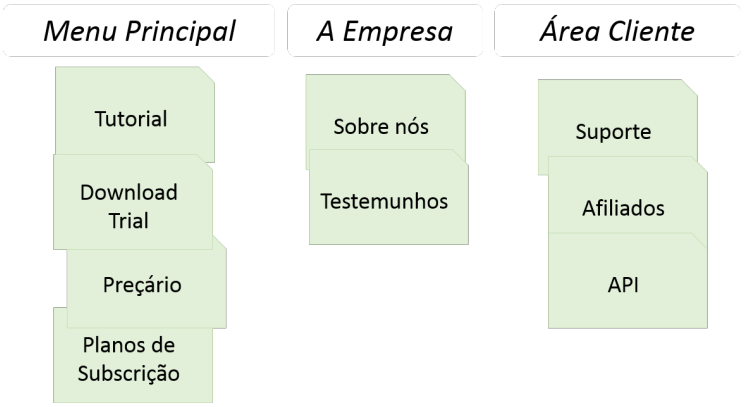


Figura 2.8: Exemplo de *Card-Sorting Aberto* [Kni, adaptado de]

Prototipagem é mais uma metodologia que visa o desenvolvimento de *design* centrado no utilizador, ou seja, que tenta que o desenvolvimento da plataforma seja feita com o acompanhamento do utilizador. Ribeiro [Rib13] define protótipo como a representação visual da interface do utilizador, podendo ser ou não interativo numa fase inicial. Salienta ainda que os protótipos não têm que replicar todas as funcionalidades do sistema, mas devem replicar integralmente tudo o que é necessário para um determinado teste.

Em suma, o uso deste tipo de metodologias, durante o processo de desenvolvimento de interfaces, é essencial, para que seja possível satisfazer de forma mais eficiente as necessidades dos utilizadores.

Capítulo 3

Especificação e Arquitetura da Plataforma DMT

Neste capítulo descrevem-se detalhadamente as fases de especificação da plataforma DMT, apresentando e explicando as opções tomadas. Com intuito de tornar a plataforma o mais funcional possível para que responda da melhor forma às necessidades do utilizador final, procurou-se de forma proativa, frequente e sistemática obter *feedback* de profissionais e outros possíveis utilizadores durante todas as fases de desenvolvimento da plataforma.

O projeto seguiu os parâmetros tradicionais de desenvolvimento de um projeto de engenharia de *software*, tendo sido orientado a partir das fases de: conceção, desenvolvimento e testes com o utilizador final, como é possível observar no esquema da Figura 3.1.

A fase de conceção dividiu-se em duas partes: levantamento de requisitos e desenho da arquitetura. O levantamento de requisitos foi feito com o auxílio de profissionais da área de data mining (DM) e está descrito na secção 3.1. Depois este levantamento são especificados os requisito na 3.2. O desenho da arquitetura, especificado no Capítulo 3.3, foi elaborado após o especificação dos requisitos.

A fase de desenvolvimento dividiu-se em duas partes, descritas no Capítulo 4 criação dos *mockups* e a implementação do protótipo funcional.

Por último, a fase de testes decorreu na parte final do projeto, após a sua implementação.

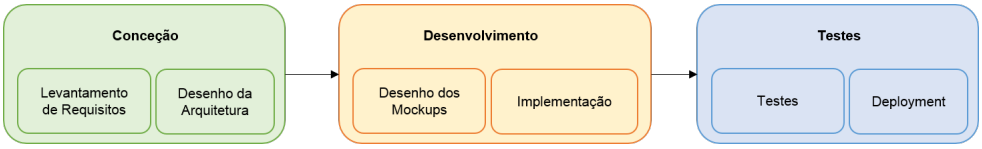


Figura 3.1: Metodologia de Desenvolvimento da Plataforma DMT

3.1 Levantamento de Requisitos

Para o levantamento de requisitos recorreu-se a informação provenientes de especialistas e profissionais da área da DM. Segue-se a descrição da metodologia associada à recolha de dados e análise dos mesmos.

3.1.1 Metodologia de Recolha de Dados

O levantamento de requisitos que permitiram a construção e adequação da plataforma DMT aos objetivos do projeto foi feito a partir da recolha de dados efetuada por inquérito. Este assumiu duas formas distintas: entrevista e questionário.

3.1.1.1 Entrevistas

Para a realização das entrevistas foi construido um guião (Anexo A.1), tendo em conta os objetivos pretendidos. Após a construção do guião foram selecionadas quatro pessoas, atendendo à sua experiência na área de DM. A entrevista foi gravada e posteriormente transcrita (Anexo A.2). Da análise e do tratamento dos dados das entrevistas foi possível construir três *Personas*. Segue-se a caracterização de cada uma delas.

Persona 1 (Persona Primária)

João Silva, de 42 anos, é Doutorado em Ciencia de Computadores, atualmente é professor associado numa faculdade de Engenharia e investigador. João trabalha frequentemente em projetos, em equipa, de data mining e usa as ferramentas R, RapidMiner. Sente como principais dificuldades no seu trabalho a troca de dados, pela inexistência de um repositório comum, que as metodologias de DM são muitas vezes desadequadas para os projetos e a dificuldade na partilha de *know-how*.

Persona 2 (Persona Secundária)

Maria Miranda, de 36 anos, é Doutorada em em Ciência de Computadores, atualmente é professora auxiliar convidada numa faculdade de Ciências e investigadora na área de data mining. Frequentemente usa as ferramentas R e Knime para analise de dados. Maria indica que a sua maior dificuldade no desenvolvimento de projetos de data mining é fazer análise de dados e implementação de algoritmos sem bases de programação.

3.1.1.2 Questionários

Os principais objetivos do questionário (Anexo A.3) foram identificar a perceção dos profissionais de Data Mining sobre as dificuldades sentidas na dinâmica entre os membros das equipas de trabalho no processo de Data Mining, identificar as funcionalidades prioritárias numa plataforma

Persona 3 (Persona Complementar)

Artur Pinto, tem 44 anos, é Doutorado em Ciências da Engenharia
e atualmente é professor numa faculdade de Engenharia. Trabalha em projetos muitos distintos nas áreas de Data Mining, Business Intelligence e Big Data usando ferramentas muito distintas em cada uma delas. A maior dificuldade que sente no seu trabalho é a falta de um formato específico para guardar os resultados das ferramentas de data mining.

colaborativa para projetos de Data Mining e ainda perceber quais as ferramentas de análise de dados mais usadas.

A amostra foi constituída por 18 indivíduos, sendo a idade mínima 22 anos, a idade máxima 44 anos, a média 29 anos e a mediana 28 anos. 83% dos indivíduos são do género masculino e 17% do género feminino. A maioria dos indivíduos da amostra tinha como habilitações académicas mestrado, maioritariamente nas áreas da Engenharia, sendo a maioria de Engenharia Informática.

O questionário está dividido em três grupos: trabalho na área de DM, ferramentas de DM e plataforma colaborativa para projetos de DM. Segue-se a análise das respostas das questões em cada um dos grupos.

- **Trabalho na área de Data Mining**

  Dos 18 indivíduos da amostra a maioria trabalha na área de DM e desses mais de 85% trabalham em equipa. Das dificuldades referidas, pelos que trabalham em equipa destacam-se: a partilha de *know-how* e a partilha de dados, tal como se apresenta na tabela 3.1.

Questão	Sim
Trabalham na área de DM	78%
Trabalham em equipa	86%
Dificuldade no trabalho em equipa:	
- Partilha de know-how	50%
- Partilha de dados	17%

Tabela 3.1: Resultados 1º grupo do questionário: Trabalho na área de DM

- **Ferramentas de Data Mining**

  Foi analisado o nível de conhecimento dos indivíduos da amostra das seguintes ferramentas: Knime, Python, R, Rapidminer e Weka. Das ferramentas analisadas constatou-se que as ferramentas mais utilizadas são: R e Rapidminer (tabela 3.2).

  Ainda neste grupo, foram objetos de análise os tipos de formatos com que os indivíduos costumam trabalhar. Atendendo à liberdade de resposta, ou seja à possibilidade de escolher mais do que uma opção, os 18 indivíduos deram origem a 41 respostas, das quais se salientam: dados relacionais e texto (tabela 3.3).

- **Plataforma Colaborativa para Projetos de Data Mining**

  Da análise da questão

Ferramenta	Conhecimento
R	78%
RapidMiner	67%
Python	50%
Weka	50%
Knime	11%

Tabela 3.2: Resultados do 2º grupo do questionário: Ferramentas de DM mais conhecidas e utilizadas

Formatos de dados	Utilização
Dados Relacionais	41%
Texto	34%
Outros	25%

Tabela 3.3: Resultados do 2º grupo do questionário: Formatos mais usados

> "Pensando numa Plataforma Colaborativa para Projetos de Data Mining, com interface que corre no browser, qual é a forma mais intuitiva de organização dos dados? "

salienta-se que 44% dos indivíduos considera que a organização deve ser *orientada aos dados*, ou seja a informação deve ser apresentada agrupando os dados disponíveis. Das funcionalidades avaliadas destacam-se as que foram consideradas com maior utilidade (útil e muito útil) e apresentam-se na tabela 3.4.

Funcionalidades	%
Guardar o nome do responsável e uma descrição textual das operações executadas nos dados;	94
Guardar os processos executados nos dados, para poderem ser usados, futuramente, noutro conjunto de dados;	89
Fazer download/upload dos ficheiros da/para a plataforma;	78
Aceder ao repositório dos dados diretamente das ferramentas de Data Mining;	78

Tabela 3.4: Resultados do 3º grupo do questionário: Funcionalidades com maior utilidade

Relativamente ao modo de utilização dos dados da plataforma constatou-se que a maioria (61%) dos indivíduos da amostra considera mais apropriado "fazer uso de uma cópia local dos dados"do que "fazer uso direto dos dados, na plataforma".

3.1.2 Resultados obtidos

Em síntese, através do resultados obtidos foi possível comprovar que a troca de dados é realmente uma das dificuldades mais sentidas pelos profissionais de DM que trabalham em equipa; as ferramentas mais usadas para análise de dados são R e RapidMiner e por isso decidiu-se que estas

seriam as duas ferramentas a ser integradas na plataforma, em primeiro lugar; por fim, tendo por base as respostas obtidas, decidiu-se que a informação seria organizada de forma orientada aos conjuntos de dados, referentes a um projeto.

3.2 Especificação dos Requisitos

Nesta secção é apresentada a especificação dos requisitos, fazendo distinção entre requisitos funcionais e requisitos não-funcionais. Os requisitos funcionais englobam o conteúdo e as funcionalidades do sistema e os requisitos não-funcionais estão associados à qualidade do software, nomeadamente, ao seu desempenho, usabilidade, fiabilidade, segurança e disponibilidade.

3.2.1 Requisitos Funcionais

Após o tratamento dos dados obtidos por questionário e a conjugação das opiniões obtidas por entrevista, consideram-se como principais requisitos funcionais, para o desenvolvimento da plataforma DMT, subdivididos em dois grupos, a interface do utilizador e as funcionalidades da plataforma.

- **Interface do utilizador:**

  A interface para o utilizador é uma das componentes deste projeto à qual se deu particular ênfase. A especificação dos seus requisitos é essencial para se conseguir uma interface dinâmica e de fácil utilização para assim atingir os objetivos definidos. O resultado é apresentado na tabela 3.5.

Requisitos da Interface	
IU001	Mostrar ficheiros de dados
IU002	Mostrar descendência entre ficheiros
IU003	Mostrar descrição dos ficheiros
IU004	Mostrar descrição das alterações feitas ao ficheiro original
IU005	Permitir ligação de um ficheiro já existente a um novo ficheiro
IU006	*Upload* de ficheiros
IU007	*Download* dos ficheiros em diferentes formatos
IU009	Selecionar ficheiro para *download*
IU010	Ver histórico de atividades
IU011	Disponibilizar fórum de discussão

Tabela 3.5: Requisitos da Interface

- **Funcionalidades da plataforma DMT:**

  Os requisitos das funcionalidades da plataforma DMT são apresentados na tabela 3.6.

Requisitos das Funcionalidades da DMT	
FP001	Fazer upload de ficheiros.
FP002	Fazer download de ficheiros.
FP003	Fazer download de ficheiros para o RapidMiner.
FP004	Fazer download de ficheiros para o R.
FP005	Fazer download de uma amostra do ficheiro.

Tabela 3.6: Requisitos das Funcionalidades da DMT

3.2.2 Requisitos Não-Funcionais

Tendo em conta a qualidade do *software* pretendida para este projeto, destacam-se como principais requisitos não-funcionais: usabilidade, escalabilidade e extensibilidade. De modo a respeitar estes requisitos foram tomadas algumas decisões relativamente às tecnologias a usar e ao padrão de arquitetura do sistema a adotar, foi ainda dada especial atenção ao desenho da interface.

3.3 Arquitetura da Plataforma DMT

A arquitetura de um sistema de *software* tem assumido ao longo dos anos um papel fundamental, no processo de desenvolvimento de *software*. Associados à definição de arquitetura de *software* surgem os conceitos de: padrões de arquitetura e estilos de arquitetura. Segundo Buschmann [BMR+96] um padrão de arquitetura expressa um conjunto de decisões de arquitetura que são aplicáveis a um problema recorrente. Os estilos de arquitetura relacionam-se com problemas do sistema relativos à especificação da estrutura geral do sistema, isto é, à sua organização, às regras de comunicação entre componentes e ao acesso aos dados. Vários autores, entre os quais Krafzing, Banke e Slama [KBS05], explicam que a arquitetura de *software* define em termos computacionais quais são os seus elementos arquiteturais e como ocorre a interação entre eles envolvendo a descrição dos mesmos e padrões que guiam a composição e as restrições sobre estes padrões.

Neste capítulo é definida a arquitetura da plataforma, discriminando os padrões e estilos de arquitetura, bem como a arquitetura física, lógica e tecnológica aplicados.

3.3.1 Estilos e Padrões de Arquitetura

Para o desenvolvimento da DMT adotaram-se como principais estilos de arquitetura: a arquitetura baseada em dados (repositório) e a arquitetura de separação lógica de processos em camadas, especificamente o padrão Model-View-Controller(MVC). Em primeiro lugar, o estilo de arquitetura que melhor se adapta ao DMT é um estilo de arquitetura baseada em dados, especificamente do tipo repositório. Este estilo de arquitetura carateriza-se por ser um sistema em que o armazenamento de dados é centralizado e comunica com vários clientes. Neste caso, trata-se de um repositório passivo, o que significa que os dados podem ser armazenados num ficheiro.

Em segundo lugar, o outro estilo de arquitetura presente na DMT é a separação lógica de processos em camadas (*Layered Architecture*). Assim a DMT encontra-se dividida em três camadas: camada de dados, camada da aplicação e camada da apresentação. A camada de dados faz ligação à base de dados, permitindo o acesso aos dados e consequente inserção e extração dos mesmos, mantendo-os independentes da lógica de negócio da aplicação. A camada da aplicação faz a ligação entre a camada dos dados e a camada de apresentação, sendo responsável pelo processamento dos dados que mantêm o estado da aplicação. Por fim, a camada de apresentação é a camada que disponibiliza a interface ao utilizador que lhe permite interagir com a plataforma. Este tipo de padrão de arquitetura confere ao sistema uma maior escalabilidade, porque permite a fácil adição de novas funcionalidades à plataforma, uma vez que as alterações feitas em qualquer uma das camadas não interferem com o funcionamento das restantes.

3.3.2 Especificação e Desenho da Arquitetura

Para facilitar e agilizar o processo de desenvolvimento e implementação, assim como para satisfazer os estilos de arquitetura escolhidos para a plataforma DMT, foram analisadas algumas *frameworks open source* de desenvolvimento de aplicações *web* disponíveis no mercado, que se adequassem aos padrões de arquitetura referidos, tais como *Ruby on Rails*, *Node.js* e *Symfony2*, sendo adotada a *framework Django* [1].

Django é uma *framework* de desenvolvimento *web* de alto nível, escrita em *Python*[2] que segue um padrão de arquitetura Model-View-Controller (MVC) e por isso vai de encontro ao padrão de arquitetura de separação lógica de processos em camadas, adotado para a criação da plataforma DMT. *Django* segue ainda o padrão *Object-Relation Mapper* (ORM) que permite que os dados sejam modelados através de classes em *Python* (*Models*), bem como a geração e manipulação das tabelas na base de dados, sem a necessidade de recorrer explicitamente a SQL. O *Django* está provido de um sistema de processamento de pedidos com um sistema de *templates web* (*View*) desenvolvidos em HTML, e faz uso de *URL* definidos por expressões regulares (*Controller*).

Aquando da criação de um novo projeto, o *Django* gera automaticamente um conjunto de funcionalidades básicas, nomeadamente uma interface dinâmica para a administração dos modelos, sistema de autenticação e operações CRUD[3] para gestão de utilizadores. *Django* é compatível com várias bases de dados relacionais.

A Figura 3.2 mostra o diagrama da base de dados usada para este sistema. É uma base de dados composta pelas tabelas: Project, User, Document, Link e Transfer.

- **User**: guarda a informação referente a utilizadores;
- **Document**: guarda a informação dos ficheiros que estão associados a cada projeto, tendo como campos, para além do nome e do conteúdo, uma descrição sobre o conteúdo do mesmo;

[1] https://www.djangoproject.com/
[2] https://www.python.org/
[3] CRUD: acrónimo Create, Read, Update, Delete: operações básicas como criar, ler, atualizar e apagar registos de uma tabela.

- **Transfer**: guarda a informação das transferências de cada ficheiro, ou seja, guarda o identificador do ficheiro transferido, o utilizador que fez a transferência, bem como o tipo de transferência: *download* ou *upload*;
- **Link**: guarda uma descrição textual das alterações feitas ao ficheiro inicial;
- **Project**: guarda a equipa de membros do projeto, ou seja, a associação de um grupo de utilizadores a um determinado projeto, o grafo de dados desse projeto e todos os ficheiros que lhe estão associados.

3.3.3 Arquitetura Física

Nesta secção apresenta-se a definição da estrutura física do sistema de *software* que está a ser desenvolvido, enfatizando as principais componentes da aplicação, a forma como estão organizadas e as dependências entre elas.

A Figura 3.3 representa a arquitetura física do sistema, ou seja, as interligações entre os vários componentes físicos. A plataforma apresenta uma arquitetura física constituída pela máquina do utilizador, servidor *web* e servidor local. O servidor local é responsável pela gestão da base de dados, assim como das funcionalidades da DMT. O servidor *web* disponibiliza a interface da plataforma à qual o utilizador pode aceder através da sua máquina.

3.3.4 Arquitetura Lógica

A arquitetura lógica da plataforma DMT, esquematizada na Figura 3.4, segue, como já referido na secção 3.3.1, um padrão de arquitetura de separação lógica de processos em camadas, especificamente o padrão MVC. Resumidamente, a arquitetura está dividida em camadas lógicas distintas, sendo a camada *View* aquela que suporta a interface para todas as funcionalidades da plataforma.

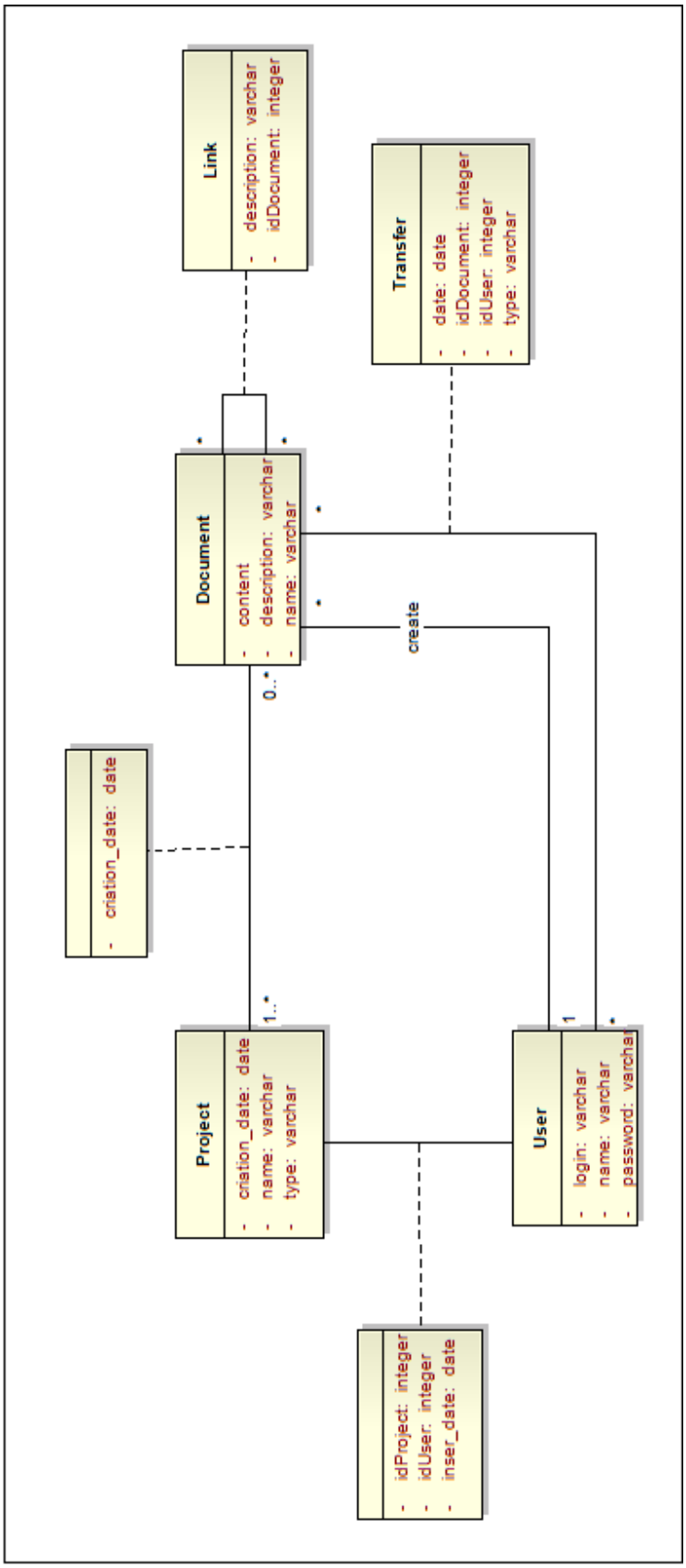


Figura 3.2: Diagrama UML da Base de Dados da plataforma DMT

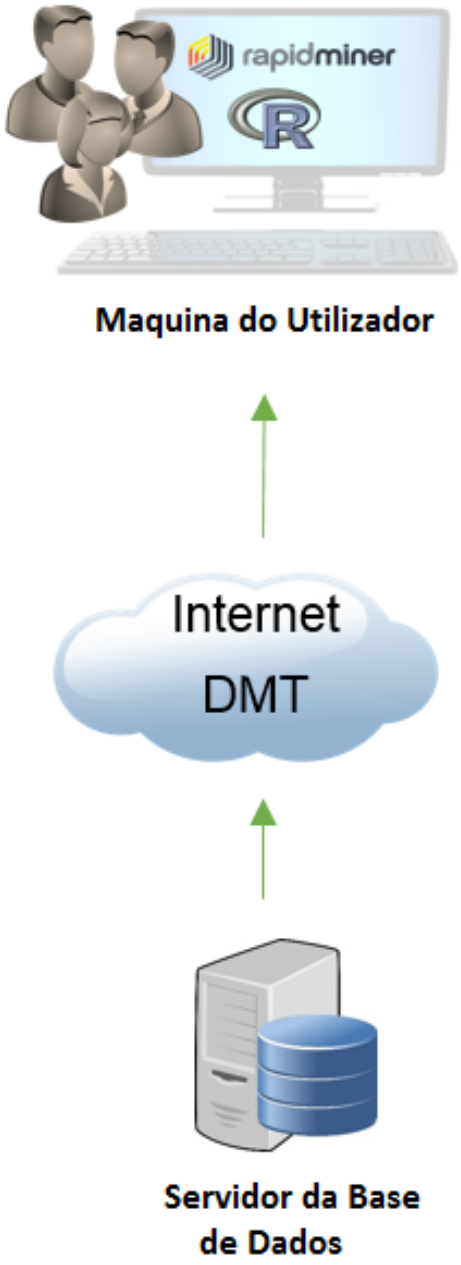


Figura 3.3: Arquitetura Física DMT

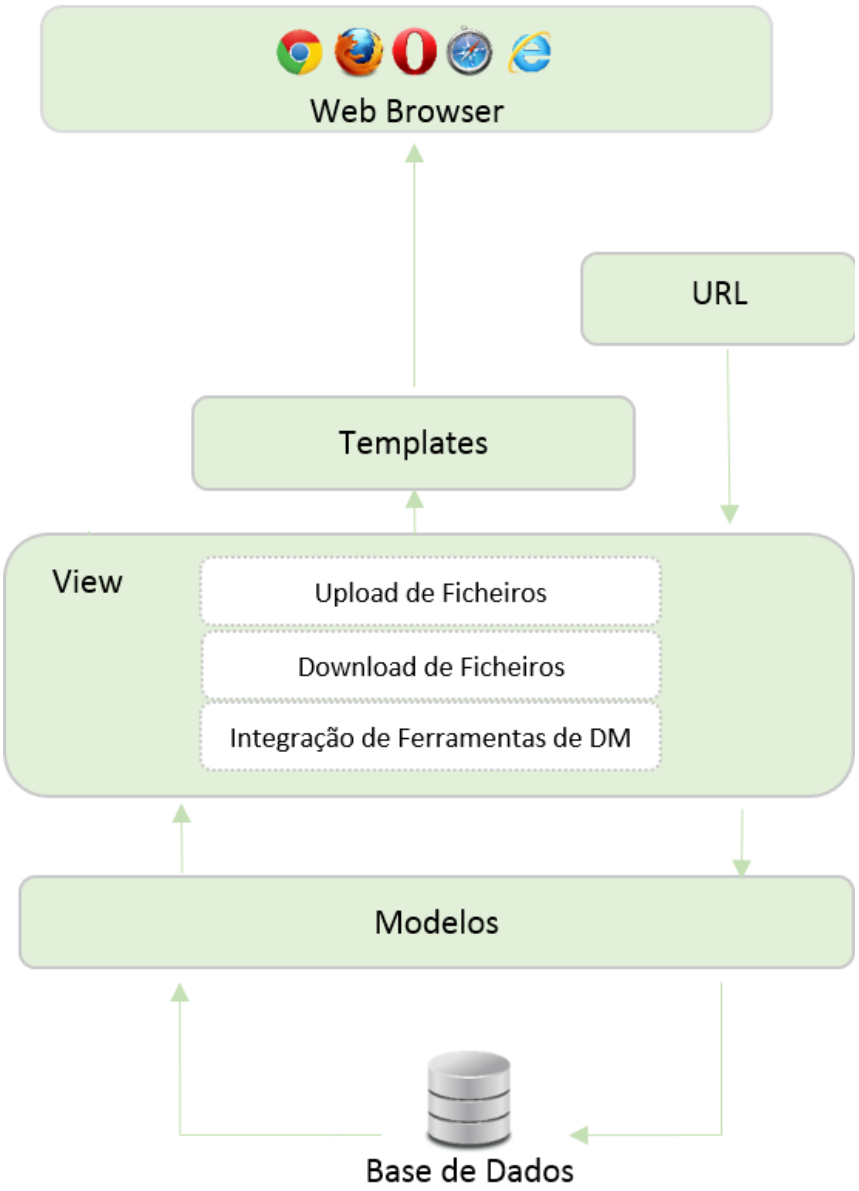


Figura 3.4: Arquitetura Lógica da plataforma DMT

3.3.5 Arquitetura Tecnológica

Na Figura 3.5 são apresentadas as tecnologias usadas no desenvolvimento da plataforma DMT. A plataforma DMT foi desenvolvida fazendo uso da *framework Django*. Como comprovado pelo estudo feito por Askins e Green [AG07] Django e Rails foram desenvolvidas para fins muito semelhantes tendo por isso uma arquitetura idêntica e não sendo claras as vantagens técnicas da utilização de uma relativamente à outra. Desta forma, a escolha passa pelo nível de conhecimento da linguagem de implementação que o programador tem, ou seja, a escolha deve ser feita tendo em conta se quem vai desenvolver se sente mais à vontade com a linguagem *Python*, usando assim *Django* ou se tem mais conhecimentos em *Ruby* e aí opta por *Rails*. É importante, também, ter em conta possíveis limitações de integração com outras componentes do sistema já existentes, o que não se aplica neste caso, uma vez que a plataforma foi implementada de raiz.

Como mencionado anteriormente, *Django* é compatível com várias bases de dados relacionais, nomeadamente *MySQL*, *SQLite3*, *PostgreSQL* e *Oracle*. Neste projeto optou-se por usar PostgreSQL pela sua escalabilidade, robustez e por não ter custos de licenciamento.

No desenvolvimento da camada de interface para o utilizador foram usadas as tecnologias HTML e CSS. A camada de lógica do negócio foi implementada em *Python*, dada a maior facilidade de integração no *Django*, também desenvolvido em *Python*.

Figura 3.5: Arquitetura Tecnológica da plataforma DMT

3.3.6 Síntese

Neste capítulo foram apresentadas as fases de levantamento e especificação de requisitos, bem como a arquitetura da DMT.

O levantamento de requisitos teve como suporte a realização de entrevistas a especialistas em DM e questionários a pessoas que trabalham na área do DM. Este levantamento de requisitos comprovou as dificuldades sentidas pelos profissionais de DM relativamente à troca de dados. Este estudo ajudou, também, na escolha das ferramentas de análise de dados a integrar na plataforma: as ferramentas R e RapidMiner.

Quanto à arquitetura foram especificados os aspetos associados à arquitetura física, arquitetura lógica e arquitetura tecnológica, bem como estilos e padrões de arquitetura. Definindo-se como principal padrão de arquitetura o padrão MVC, uma especificação do padrão de arquitetura separação lógica de processos em camadas. Foi também identificada a *framework Django* que serve de base para a plataforma, sobre a qual assentou o protótipo funcional desenvolvido, descrito no capítulo seguinte.

Capítulo 4

Interface e Implementação da Plataforma DMT

Neste capítulo é mostrada a forma como a plataforma DMT, e em particular a sua interface, foram inicialmente projetada através de *mockups*, apresentando-se de seguida o ambiente de desenvolvimento do projeto e finalmente são especificadas as funcionalidades que foram efetivamente implementadas, explicando o protótipo funcional desenvolvido.

4.1 Desenho da Interface

O processo de desenho da interface iniciou-se com a construção de alguns esboços (*mockups*) que ilustram a interface para o utilizador da plataforma DMT, tendo em consideração todo o estudo feito anteriormente. Os *mockups* da plataforma foram criados recorrendo à ferramenta *Balsamiq Mockups* para *Google Drive*[1].

Com base no levantamento de requisitos, referido no capítulo 3, decidiu-se que a organização da informação na plataforma DMT seria feita por projetos, os quais teriam uma organização orientada aos dados. Para facilitar o manuseamento dos dados de cada projeto optou-se por usar grafos, estrutura que permite facilmente identificar relações entre os mesmos.

A Figura 4.1 é o esboço desenhado para a página de um projeto na plataforma DMT.

Cada ficheiro de dados corresponde a um nó (Figura 4.1 A) e as arestas guardam uma descrição textual (Figura 4.1 B) sobre as transformações feitas ao "ficheiro-pai", isto é, alterações feitas ao ficheiro de dados que lhe deu origem. As ligações devem ser setas unidirecionais para que o utilizador perceba facilmente quais os ficheiros que são resultado da alteração de outros ficheiros.

Foi desenhado um botão (Figura 4.1 C) para fazer o *upload* dos ficheiros de dados na plataforma. Depois de escolhido o ficheiro e de adicionado ao projeto é pedido ao utilizador que insira a descrição do mesmo, como mostra a Figura 4.2.

Aquando de um *click* no ficheiro pretendido é mostrado um menu com as opções de *download*, como é possível ver na Figura 4.3, bem como as opções de: edição da descrição e eliminação do

[1] https://balsamiqgdrive.appspot.com

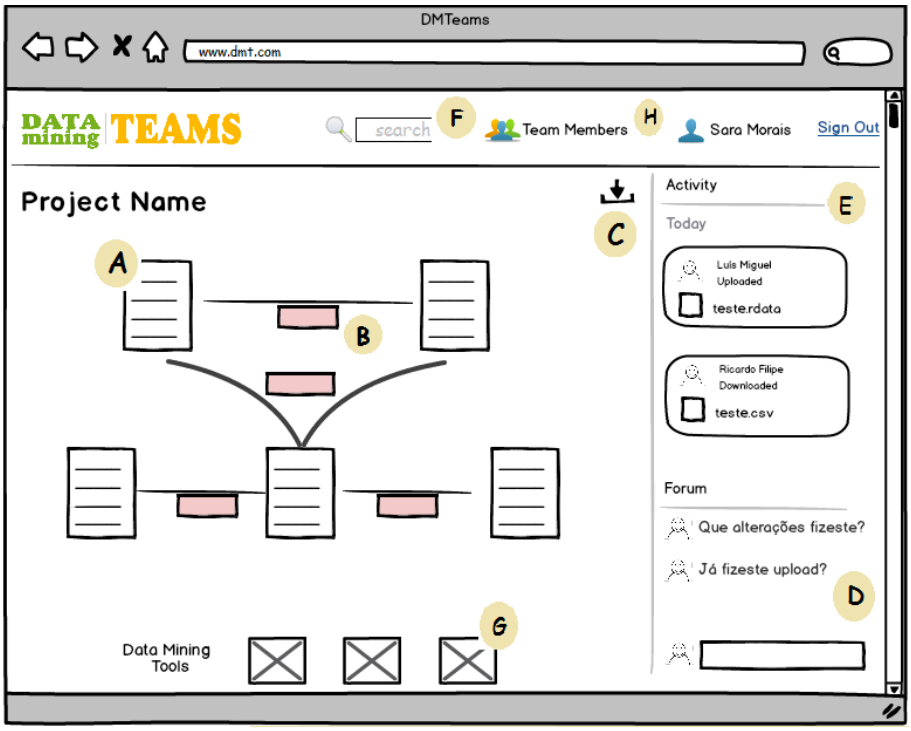


Figura 4.1: Página de um Projeto na plataforma DMT: A) Ficheiro; B) Descrição da Ligação; C) Botão de Upload; D) Fórum de discussão; E) Registo de Atividade do Projeto; F) Motor de Pesquisa; G) Logótipo das Ferramentas de DM H) Elementos da Equipa;

ficheiro. As opções de *download* permitem ao utilizador escolher qual das ferramentas de análise de dados deseja usar.

O acesso aos projetos na DMT é feito depois do utilizador se registar e autenticar na plataforma DMT. Quando o utilizador se autentica tem acesso a todos os projetos em que está envolvido e é-lhe dada a possibilidade de criação de um novo projeto como apresenta a Figura 4.4.

Como se trata de uma plataforma colaborativa, para promover uma melhor comunicação entre os membros das equipas, idealizou-se que a DMT seria provida de um fórum de discussão, representado Figura 4.1 D e ainda apresentada a informação referente à atividade dos vários utilizadores no projeto, como se pode ver na Figura 4.1 E. Cada utilizador teria a possibilidade de ver os membros da equipa (Figura 4.1 H) e fazer pesquisas pelo nome do ficheiro (Figura 4.1 F).

Projetou-se que as ferramentas de análise de dados estariam no fundo da página, representadas pelo respetivo logótipo, assim o utilizador apenas tinha que arrastar o ficheiro para aquele que desejasse usar e na qual pretendia editar os seus ficheiros.

4.2 Ambiente de Desenvolvimento

Ao longo do projeto foram usadas as seguintes ferramentas de auxílio ao desenvolvimento:

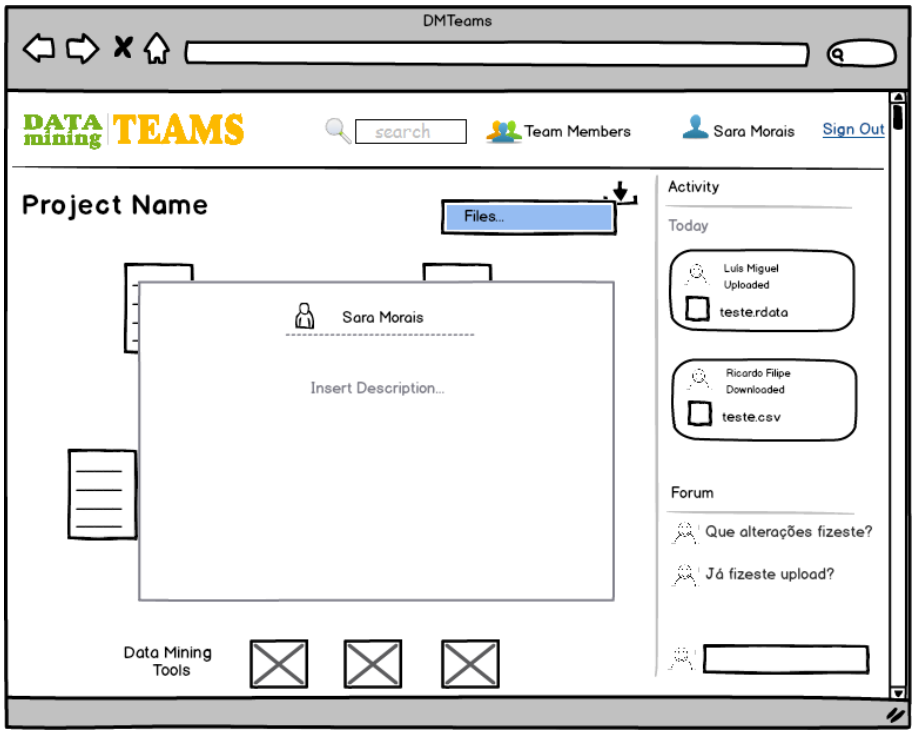


Figura 4.2: Janela de Descrição da Ligação

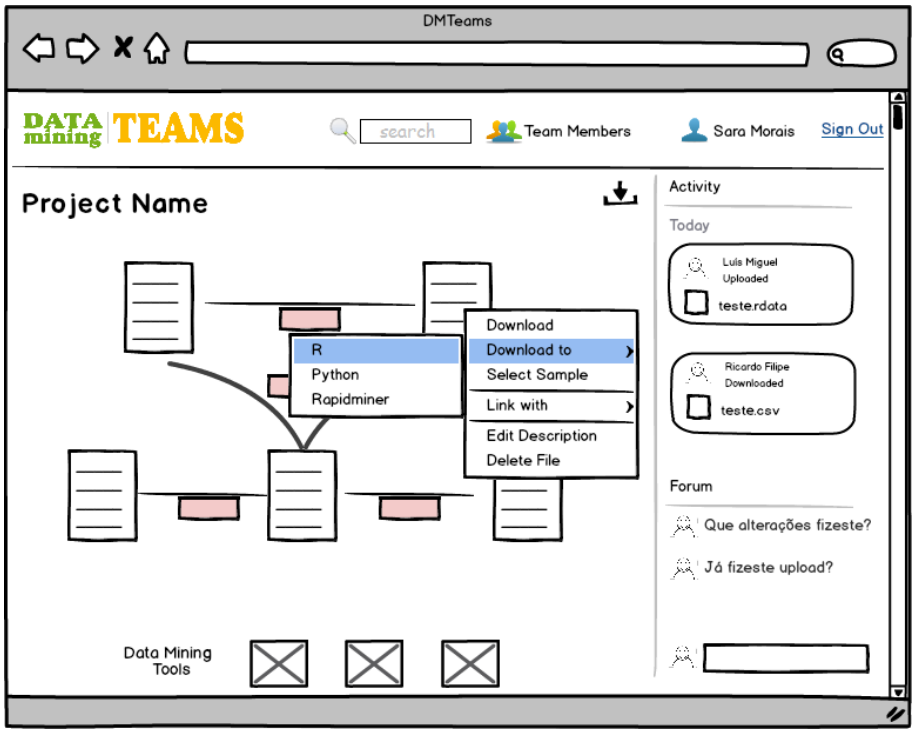


Figura 4.3: Menu de Download de Ficheiros

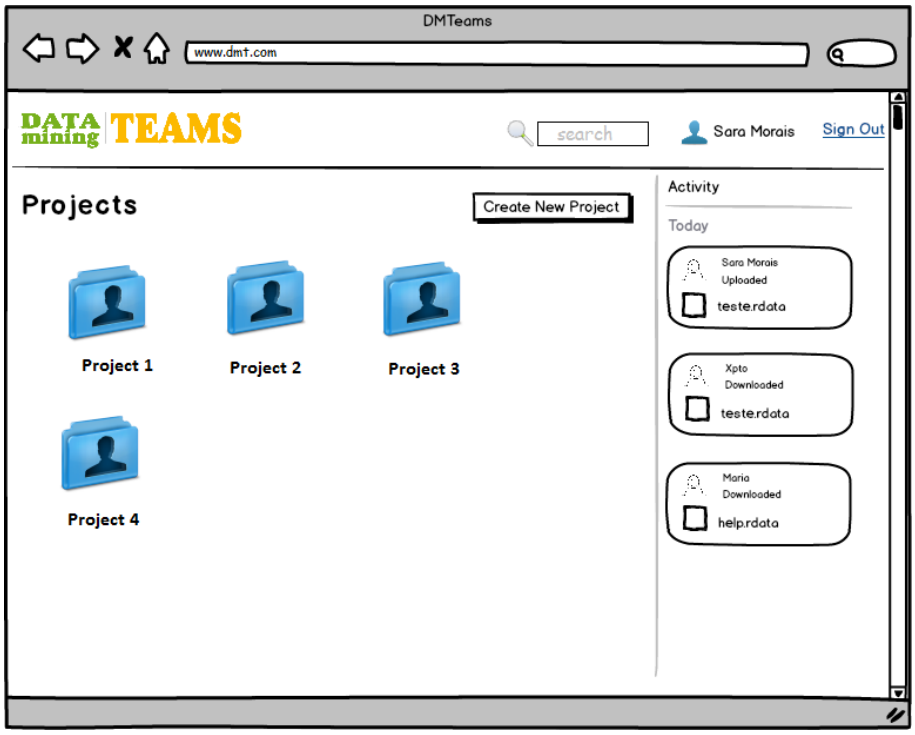


Figura 4.4: Página dos Projetos do Utilizador

- **Servidor Virtual** instalado e configurado sob o domínio da FEUP no sistema operativo Linux 3.2.5.7 (Debian);
- **Google Drive** usado para armazenamento de toda a documentação gerada.
- **Aptana Studio 3** foi o IDE usado para a criação do código para a plataforma DMT, com uma ligação à maquina virtual;
- **Github** para controlo de versões do código da plataforma.

4.3 Protótipo Funcional

Pode considerar-se que a implementação da plataforma é o culminar do projeto, no sentido de o tornar visível e com sentido prático as várias etapas que constituem este trabalho. Para a criação do protótipo funcional, e dadas as limitações temporais, foi identificado um subconjunto de funcionalidades consideradas prioritárias a incluir no mesmo.

Com foco no objetivo objetivo do projeto associado ao manuseamento e utilização de grandes quantidades de informação provenientes de diversas fontes e em diversos formatos, foram implementadas as seguintes funcionalidades:

- *Upload* de ficheiros e inserção da respetiva descrição;
- *Download* de ficheiros em formatos compatíveis com a ferramenta *R*;

- *Download* de ficheiros em formatos compatíveis com a ferramenta *Rapidminer*;
- Criação de ligação com descrição entre os ficheiros inseridos na DMT.

Das ferramentas de DM, estudadas anteriormente, o *R* e o *Rapidminer* foram as adotadas para integração nesta fase do projeto. Esta escolha foi fundamentada, pelas respostas obtidas pelo estudo feito durante o levantamento de requisitos (detalhado no capítulo 3) e impulsionada pelo facto de se querer integrar, nesta fase, uma ferramenta de análise de dados com interface gráfica para o utilizador e outra cuja utilização é feita por linha de comandos.

De seguida as secções 4.3.1, 4.3.2 e 4.3.3 permitem compreender melhor como foi implementado cada um dos módulos da plataforma DMT.

4.3.1 Modelos

Para a criação da base de dados, apresentada na secção 3.3.2, foi criado um Modelo, escrito em *Python*, para cada uma das tabelas. Na listagem 4.1 é um exemplo desses modelos, especificamente o modelo correspondente à tabela *Document* que guarda a informação dos ficheiros que são inseridos na DMT.

```
1 class Document(models.Model):
2     nodeId = models.IntegerField()
3     name = models.CharField(max_length=50)
4     description = models.CharField(max_length=400)
5     docfile = models.FileField(upload_to='documents/%Y/%m/%d')
6     userD = models.ManyToManyField(User)
7
8     def __unicode__(self):
9         return self.name
```

Listing 4.1: Extrato de código *Python*: Definição do modelo *Documents*

No total foram definidos oito modelos: um modelo por cada tabela da base de dados definida para a plataforma, apresentada na secção 3.3.2 e um modelo adicional, o modelo *"LastGraphJson"*, apresentado na listagem 4.2. Este modelo é responsável por guardar na base de dados a estrutura atual do grafo e facilitar a sua sincronização entre o servidor e o *browser*.

```
1 class LastGraphJson(models.Model):
2     version_id = models.AutoField(primary_key=True)
3     last_graph_json = models.TextField()
```

Listing 4.2: Extrato de código *Python*: Definição do modelo *LastGraphJson*

4.3.2 Views

As *views* onde é definida a lógica da plataforma, também foram desenvolvidas em *Python*, como é possível ver na listagem 4.3, onde é definida a *view* que permite fazer o *download* de ficheiros com a extensão *.RData*.

```
1  def downloadR(request, docPath=''):
2
3      import os, tempfile, zipfile
4      from django.core.servers.basehttp import FileWrapper
5      from django.conf import settings
6      import mimetypes
7
8      docPathWithoutCsv = os.path.splitext(docPath)[0]
9
10     subprocess.call(['Rscript', "web_dmt/DMT-R convert csv to RData.R",
           docPathWithoutCsv])
11
12     filename     = docPathWithoutCsv + ".RData"
13     download_name = filename
14     wrapper      = FileWrapper(open(filename))
15     content_type = "application/x-www-form-urlencoded"
16     response     = HttpResponse(wrapper,content_type=content_type)
17     response['Content-Length']      = os.path.getsize(filename)
18     response['Content-Disposition'] = "attachment; filename=%s"%download_name
19     return response
```

Listing 4.3: Extrato de código *Python*: Definição da *view* para o *downoad* de ficheiros no formato *RData*

Foram implementadas as *views*: *List, Graph, Link, DocumentList, downloadCSV, downloadR, downloadRapidMiner, savelastchange* e *loadlastchange*.

- **List**: *view* responsável por instanciar o formulário para o *upload* de um novo ficheiro e por guardar esse ficheiro na base de dados;
- **Graph**: *view* responsável por carregar a página inicial da plataforma DMT;
- **Link**: *view* que tem como finalidade criar o formulário para inserir a descrição da relação entre dois ficheiros e para a guardar na base de dados;
- **DocumentList**: *view* auxiliar que lista todos os ficheiros que existem na base de dados para, posteriormente, ser possível fazer o seu *download*
- **downloadCSV**: *view* responsável pela funcionalidade de *download* de um ficheiro sem qualquer tipo de alteração ao seu formato;

- **downloadR**: *view* que tem como objetivo fazer a conversão de ficheiros *.csv* para ficheiros com o formato *.RData* e o seu *download*;
- **downloadRapidMiner**: *view* responsável pela preparação dos ficheiros para que possam ser usados fazendo uso da ferramenta RapidMiner;
- **savelastchange**: *view* que atualiza a tabela que guarda os grafos na base de dados;
- **loadlastchange**:*view* que permite que o grafo gerado até ao momento, num determinado projeto, seja carregado na DMT.

4.3.3 Implementação da Interface

Comparativamente ao desenho inicial da interface foram feitas algumas alterações durante a implementação que são descritas mais à frente. A Figura 4.5 apresenta a plataforma DMT no estado mais atual.

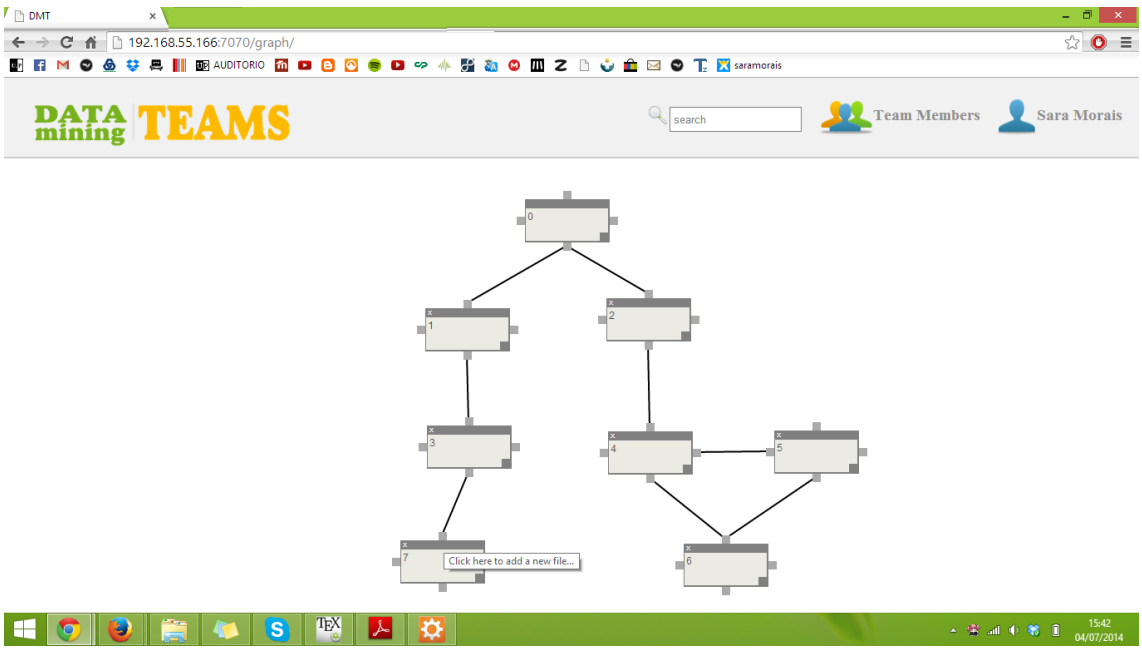


Figura 4.5: Exemplo de Utilização da Plataforma DMTs

As principais decisões de desenho implementadas estão sintetizadas na tabela 4.1.

Como referido anteriormente, optou-se por usar grafos para a organização dos ficheiros de dados na DMT, para tal, foi feita uma pesquisa para encontrar um editor de grafos *open-source* que melhor respondesse às necessidades da DMT, ou seja, que permitisse a criação de grafos, especificamente a criação e eliminação de nós e arestas. Outro critério tido em conta na procura do editor de grafos era que este estivesse implementado com tecnologia compatível com *Django*, para que fosse possível a sua integração na DMT. Foram encontradas várias possibilidades, nomeadamente: *Znode*, *GoJS*, *WireIt*, *arbor.js*. Aquela que apresentava especificações que melhor se adaptavam às

Requisitos da Interface		Interface
IU001	Mostrar ficheiros e descendência entre eles	Grafo carregado quando se entra na DMT
IU002	Mostrar descrição dos ficheiros	Abre *dialog* após clicar no ficheiro pretendido
IU003	Mostrar descrição das alterações feitas ao ficheiro original.	*Dialog* após clicar na ligação da transformação pretendida
IU004	Permitir ligação de um ficheiro já existente a um novo ficheiro.	Desenha a ligação arrastando o rato de um nó para o outro
IU005	*Upload* de ficheiros	Clicar no ecrã e abre *popup* para *upload*
IU006	*Download* dos ficheiros em diferentes formatos	Botão na *dialog* com o nome e descrição do ficheiro
IU007	Selecionar ficheiro para *download*	Clicar no nó correspondente ao ficheiro pretendido
IU008	Ver histórico de atividades.	Não implementado
IU009	Disponibilizar fórum de discussão.	Não implementado

Tabela 4.1: Principais decisões de desenho da interface

necessidades da DMT pela simplicidade do código fonte e por permitir a costumização do mesmo, foi o editor de grafos *Znode*[2]. Assim, foi incorporado na DMT o editor de grafos *Znode*. O *Znode* é um editor de grafos *open source* para *web*, que permite a visualização e organização da informação em grafos, correspondendo cada nó do grafo na plataforma DMT, a um ficheiro. O editor está escrito em *javascript* [3] fazendo uso de duas bibliotecas de suporte: *jQuery*[4] e *Raphaël*[5].

A interface gráfica faz uso de *templates* escritos em *HTML e CSS* e usando também *JavaScript*. De seguida são descritas com mais detalhe as opções de implementação, relativamente à interface, das funcionalidades de *Upload* e *Download*.

4.3.3.1 Upload

O *upload* de ficheiros na plataforma DMT é feito através de uma janela de formulário criada quando o utilizador clica na página do projeto, como mostra a Figura 4.7, dispensando a necessidade do botão pensado inicialmente, no sentido de facilitar a experiência do utilizador. Foi colocada uma *tooltip* (Figura 4.6) com o texto *"Click here to add a new file..."*, para tornar a ação de *upload* de um novo ficheiro mais percetível para o utilizador.

O formulário da descrição da relação entre dois ficheiros é semelhante ao de *upload* como se pode ver na Figura 4.8.

4.3.3.2 Download

O *download* de ficheiros da plataforma é feito atraves de um botão que existe na *dialog* aquando de um *click* num dos nós do grafo, como mostra a Figura 4.9.

[2] http://zreference.com/znode/
[3] http://www.javascriptsource.com/
[4] http://jquery.com/
[5] http://raphaeljs.com/

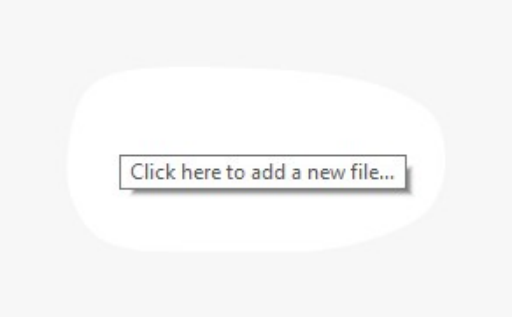


Figura 4.6: *Tooltip*

Neste primeiro protótipo funcional, uma vez que não foi possível implementar na plataforma a possibilidade de correr processos de RapidMiner, os utilizadores da ferramenta RapidMiner, antes de começarem a usar a DMT, precisam de fazer as seguintes configurações:

1. Criar um novo repositório no *RapidMiner* com o nome "DataMiningTeams-DefaultRepository";
2. Fazer *download* do processo "DMT-import.rmp"da página da DMT;
3. Importar o processo para o *RapidMiner*;
4. Criar um diretório "DMT-Config"no seu ambiente de trabalho.

Depois das configurações terminadas, quando o utilizador faz um *download* da plataforma apenas tem que correr o processo anteriormente importado para o repositório criado no *RapidMiner*. É importante que o utilizador, quando estiver a fazer *download* de um ficheiro da plataforma, mantenha o nome do ficheiro como ele é gerado pela DMT. O diagrama 4.10 esquematiza todo este processo.

Os utilizadores da ferramenta *R* não têm necessidade de configurações prévias. Quando é feito um *download*, através do botão de *download R* (Figura 4.11) é corrido um *script .sh* no servidor que faz a conversão dos ficheiro *.csv* para *.RData* antes do ficheiro ser descarregado no computador do utilizador. O diagrama da Figura 4.9 descreve a sequência de eventos despoletados para esta ação.

4.3.4 Síntese

Os aspetos enfatizados neste capítulo foram o desenho da interface, o ambiente de desenvolvimento da plataforma DMT e a criação do protótipo funcional.

O desenho da interface, partiu de um esboço inicial e foi sendo modificado ao longo da sua implementação até à fase atual, como foi apresentado na Figura 4.5.

O protótipo funcional resultante apresenta as funcionalidades definidas como mais prioritárias, permitindo assim a troca de dados entre os elementos de uma equipa e a conversão e preparação dos mesmos para que fosse possível a sua utilização nas ferramentas R e RapidMiner.

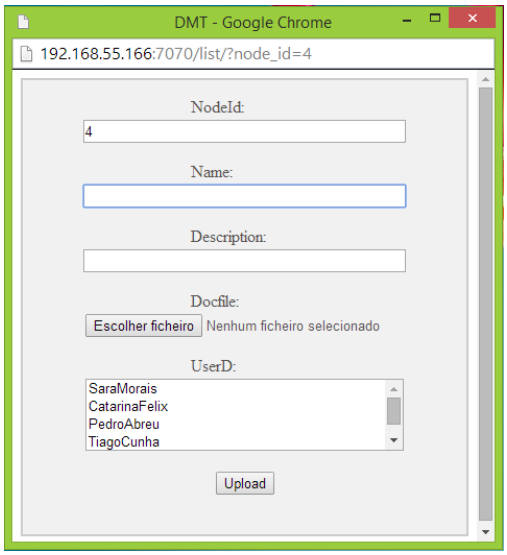


Figura 4.7: Formulário de *upload* para a DMT

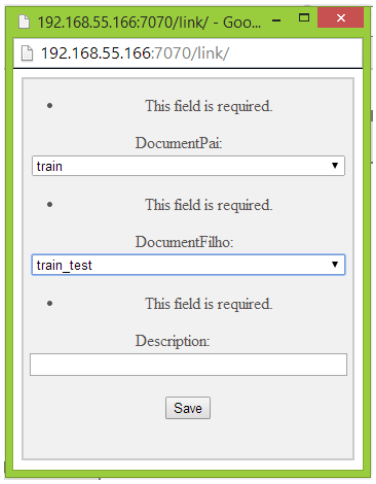


Figura 4.8: Formulário para a descrição da relação entre dois ficheiros na DMT

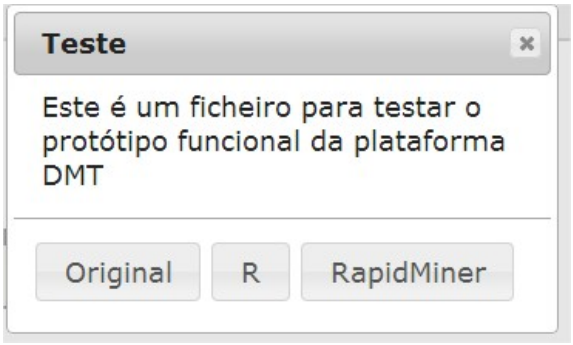


Figura 4.9: Janela para *download* de um ficheiro exemplo com o nome "Teste"

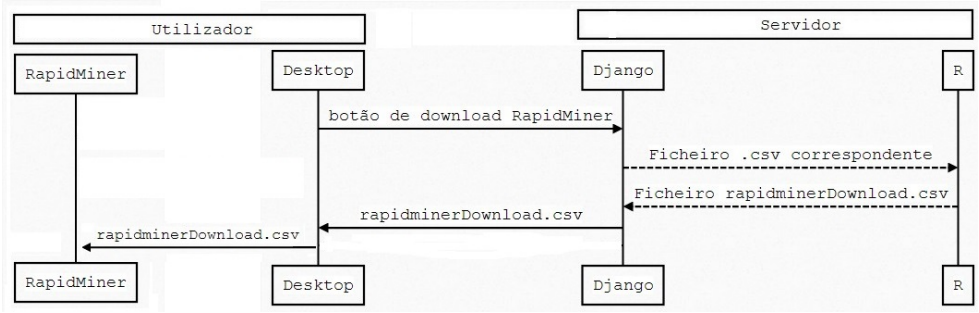


Figura 4.10: Diagrama de Sequência do *Download* de Ficheiro para RapidMiner

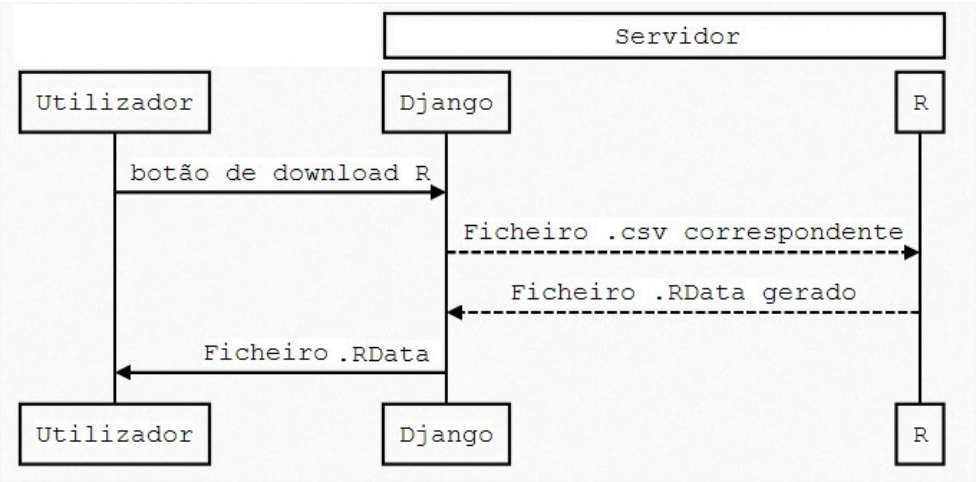


Figura 4.11: Diagrama de Sequência do *Download* de Ficheiro .RData

Capítulo 5

Testes e Resultados

Neste capítulo apresentam-se o ambiente de avaliação utilizado para os testes do protótipo funcional da plataforma DMT desenvolvida e os resultados identificados como mais relevantes.

5.1 Condições Experimentais

O processo de avaliação do protótipo funcional da plataforma DMT teve início com a seleção de uma equipa constituída por quatro pessoas, que trabalham frequentemente sobre a resolução de problemas de DM, embora tenham *backgrounds* distintos. A equipa foi dividida em dois grupos, designados por Grupo A e Grupo B (para efeitos da avaliação). Em cada grupo, um elemento usa uma ferramenta de análise de dados com interface gráfica - RapidMiner - e o outro uma ferramenta de análise de dados de linha de comandos - a ferramenta R no Grupo A, e Python no Grupo B.

Após a constituição das equipas, passou-se à definição do contexto da avaliação. Tendo em consideração o contexto onde surgiu a motivação inicial deste projeto, relacionada com a participação de equipas de profissionais de DM em competições mundiais de DM, decidiu-se que a avaliação da plataforma DMT seria baseada em dados de uma dessas competições. Assim foi decidido que a equipa participaria na competição "*Acquire Valued Shoppers Challenge*"[1], promovida pela comunidade de *data scientists Kaggle*[2]. Estipulou-se que a participação na competição teria uma duração de 3 semanas, a começar a 16 de junho de 2014 e com final a 4 de julho de 2014.

A "*Acquire Valued Shoppers Challenge*"tem como principal objetivo prever que clientes são propensos a repetir a compra do mesmo produto utilizando descontos oferecidos pelas marcas. Para o desenvolvimento da solução, a *Kaggle*, forneceu aos participantes um histórico de descontos de compras oferecidos numa campanha a um vasto conjunto de clientes. Estes dados traduzem-se numa base de dados com quase 350 milhões de linhas de dados completamente anónimos de mais de 300.000 clientes. A competição teve início no dia 10 de abril de 2014 e termina oficialmente no dia 14 de julho de 2014.

[1] https://www.kaggle.com/c/acquire-valued-shoppers-challenge
[2] https://www.kaggle.com/

Considerando como funcionalidades essenciais do protótipo desenvolvido a troca de dados entre os membros da equipa e a conversão dos mesmos, foi sobre estes dois aspetos que incidiu a avaliação, tendo cada grupo um papel distinto na avaliação. O Grupo A desenvolveu a sua solução tendo como suporte a plataforma DMT. Por sua vez, o Grupo B usou apenas as ferramentas de análise de dados. Os objetivos finais referente ao Grupo A eram:

- Saber se fez uso das funcionalidades da DMT, ou seja, saber se houve realmente troca de dados entre os dois elementos recorrendo à plataforma;
- Perceber quais as principais dificuldades sentidas e que aspetos acharam mais positivas na plataforma.

Relativamente ao Grupo B o objetivo era:

- Perceber quais as maiores dificuldades sentidas no decorrer do desenvolvimento da solução também no que diz respeito à colaboração e, em particular à troca de dados.

5.2 Resultados

Tendo em conta que a avaliação decorreu num curto período de tempo e o reduzido número de elementos da equipa, os resultados têm apenas um cariz qualitativo, baseado nas opiniões mais relevantes dos elementos da equipa usada para a avaliação. Estas opiniões foram recolhidas através de conversas informais com os quatro elementos da equipa.

O **Grupo A** salientou como aspetos positivos:

- Terem podido trocar de ficheiros de dados;
- A possibilidade de fazer *download* dos ficheiros nos formatos já compatíveis com as ferramentas de análise de dados que estavam a usar;
- A possibilidade de fazer *upload* de ficheiros em vários formatos que não só *.csv*.

E como aspetos a melhorar:

- A sensibilidade do ecrã ao *click* do rato, que levava à criação de novos nós não desejados;
- Ser possível selecionar processos para aplicar aos dados na própria DMT sem que seja necessário usar uma ferramenta de análise de dados;

O **Grupo B** apontou como principal dificuldade a interação de cada um dos elementos com os dados gerados pelo outro. Explicaram que seria interessante se fosse possível interagir com os dados um do outro tendo acesso visual ao ecrã um do outro, através de computadores distintos, permitindo assim arrastar a informação de um ecrã para o outro "diretamente", mas continuando a trabalhar cada um em seu ambiente. Estas observações reforçam a motivação por trás deste projeto.

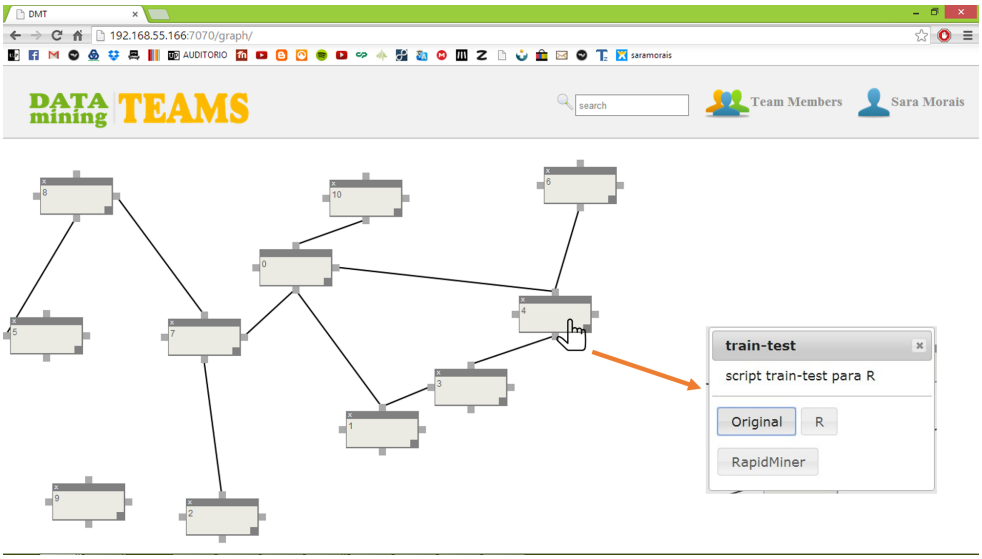


Figura 5.1: Grafo gerado pelo Grupo A durante a fase de avaliação da DMT

Na Figura 5.1 é possível ver o grafo gerado pelo Grupo A durante o período de avaliação. No total foram transferidos 11 ficheiros fazendo uso da plataforma, com diferentes formatos.

Embora os resultados não possam ser generalizados atendendo à forma como foram obtidos, pode concluir-se que o trabalho até agora desenvolvido constituiu um desafio que merece empenho e continuação por se acreditar que pode ser um bom contributo para a melhoria do desempenho das equipas na participação deste tipo de desafios.

Estes resultados foram escritos quando ainda não havia resultados das submissões feitas pela equipa na competição.

Capítulo 6

Conclusões e Trabalho Futuro

Da necessidade de trabalhar dados e resultados de diferentes modelos e formatos, no contexto de projetos de DM, resultou o desenvolvimento e implementação do projeto traduzido nesta dissertação e a concretização da plataforma Data Mining Teams (DMT).

Os principais objetivos que precederam este projeto contemplavam desenhar e construir uma plataforma colaborativa para projetos de DM com ambiente gráfico interativo e colaborativo que permitisse importar e explorar dados e resultados independentemente dos formatos dos mesmos. Sendo este um projeto bastante ambicioso, tendo em conta o tempo disponível para a sua concretização, o resultado desta dissertação não respondeu funcionalmente a todos dos requisitos especificados para o produto final. No entanto, as fases de conceção e desenho da arquitetura foram detalhadas considerando a totalidade das funcionalidades da plataforma, e foi implementado um protótipo funcional que serviu como prova de conceito para as principais funcionalidades especificadas.

A primeira fase do projeto consistiu na análise do problema proposto para esta dissertação, investigando o estado da arte para aprofundar os conhecimentos sobre o tema, levando a uma melhor compreensão da necessidade do produto, e ainda à perceção das alternativas que existem no mercado para ajudar no desenvolvimento de uma solução. O estudo das metodologias de DM permitiu perceber a clara necessidade de troca de dados e transformação destes durante o processo de DM em equipa. Com o estudo da interação pessoa-computador foi traçada a metodologia a usar para o levantamento de requisitos mais adequado tendo em vista o desenvolvimento de uma plataforma com elevado nível de usabilidade e com *design* centrado no utilizador.

A análise do problema com mais detalhe levou a concluir que a melhor solução passaria por uma plataforma disponível via *web*, para que pudesse assim satisfazer o requisito de ser colaborativa. De forma a definir com mais rigor os requisitos do produto, para que satisfizesse da melhor forma o utilizador final, foram feitas entrevistas e inquéritos a profissionais que trabalham com projetos de DM. Este levantamento de dados permitiu especificar os requisitos com mais detalhe, e indo ao encontro das necessidades reais dos utilizadores.

Para o desenvolvimento da plataforma em questão identificaram-se como principais estilos de arquitetura a arquitetura baseada em dados; e a arquitetura baseada em separação lógica de

processos em camadas, para que seja um sistema mais escalável.

Relativamente às ferramentas que foram usadas, após o estudo realizado sobre as várias alternativas possíveis, optou-se pela utilização da *framework Django* para agilizar o processo de desenvolvimento do protótipo funcional. Sempre com o objetivo de fácil utilização por parte do utilizador, decidiu-se que a informação seria apresentada em forma de grafo, para visualmente se ter uma rápida perceção dos ficheiros de dados disponíveis e respetivos ficheiros de dados transformados.

Fruto de alguns imprevistos que levaram a restrições temporais, protótipo funcional resultante apresenta as funcionalidades definidas como mais prioritárias, permitindo assim a troca de dados entre os elementos de uma equipa e a conversão e preparação dos mesmos para que fosse possível a sua utilização nas ferramentas R e RapidMiner.

Os testes e avaliação da plataforma DMT tiveram um cariz unicamente qualitativa, porque por questões de enquadramento de calendário, apenas foi possível conseguir um número reduzido de sujeitos de teste e por um curto espaço de tempo. Assim, foi disponibilizada a plataforma a uma equipa de DM que a pudesse usar durante a participação numa competição de DM. Seria necessário fazer testes mais exaustivos para uma melhor avaliação da DMT, no entanto os que foram realizados apresentaram já boas indicações e melhorias a fazer.

O trabalho realizado focou mais no levantamento de requisitos e especificação da solução. Isto implicou menos esforço para a parte de implementação e testes, tendo os resultados destes ficado um pouco aquém do desejado.

6.1 Trabalho Futuro

Tendo em conta que continuam bastante válidos os objetivos que precederam esta dissertação, é importante continuar este projeto e aproveitar o trabalho realizado até ao momento. Assim, o trabalho futuro está dividido em curto/médio e longo prazo, incluindo as funcionalidades não implementadas já definidas nos requisitos e novas sugestões.

Numa visão a curto/médio prazo seria importante refinar as funcionalidades implementadas, nomeadamente: a apresentação do grafo de forma a que fosse possível ver o nome do ficheiro correspondente a cada nó no próprio nó; e ainda tornando o grafo mais apelativo no que diz respeito à cor e ao *design*. Para além disso seria interessante implementar o sistema de autenticação de utilizadores e o *chat* na página de cada projeto, tal como foi pensado da definição dos requisitos.

Numa perspetiva a longo prazo, sugere-se o desenvolvimento da funcionalidade sugerida que possibilite ao utilizador correr processos de transformação dos dados na própria DMT, sem que tenha que fazer uso das ferramentas. De um ponto de vista ainda mais ambicioso, traria grandes vantagens que a plataforma permitisse ao utilizador, aquando da criação de um novo projeto, escolher o tipo de projeto, relativamente aos formatos dos dados a ser utilizados naquele projeto.

Referências

[AG07] Ben Askins e Alan Green. A rails / django comparison. In *The Python Papers*, volume 2 of *2*. Open Source Developer's Conference, 12 2007.

[BMR+96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad e Michael Stal. *Pattern-oriented Software Architecture: A System of Patterns*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[Bra07] Max Bramer. *Principles of Data Mining*. Springer, 2007.

[CC11] Richard Caddick e Steve Cable. *Communicating the User Experience*. John Wiley and Sons Ltd, 3rd edition, 2011.

[CCK+00] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer e Rudiger Wirth. Crisp-dm 1.0 step-by-step data mining guide. Technical report, The CRISP-DM consortium, August 2000. URL: http://www.crisp-dm.org/CRISPWP-0800.pdf.

[CHS+98] Peter Cabena, Pablo Hadjinian, Rolf Stadler, Jaap Verhees e Alessandro Zanasi. *Discovering Data Mining: From Concept to Implementation*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[Cos11] Joel Frederico Azevedo Costa. Um ambiente gráfico para facilitar tarefas de data mining via ferramenta r. Technical report, Universidade do Minh - Escola de Engenharia, Outubro 2011.

[CRC07] Alan Cooper, Robert Reimann e David Cronin. *About Face 3: The Essentials of Interaction Design*. John Wiley & Sons, Inc., New York, NY, USA, 2007.

[DFAB97] Alan Dix, Janet Finlay, Gregory Abowd e Russell Beale. *Human-computer Interaction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.

[GARM+08] P. González-Aranda, Ernestina Menasalvas Ruiz, Socorro Millán, Carlos Ruiz e Javier Segovia. Towards a methodology for data mining project development: The importance of abstraction. In Tsau Young Lin, Ying Xie, Anita Wasilewska e Churn-Jung Liau, editors, *Data Mining: Foundations and Practice*, volume 118 of *Studies in Computational Intelligence*, pages 165–178. Springer, 2008. URL: http://dblp.uni-trier.de/db/series/sci/sci118.html#Gonzalez-ArandaRMRS08.

[GG99] Michael Goebel e Le Gruenwald. A survey of data mining and knowledge discovery software tools. *SIGKDD Explorations*, 1:20–33, 1999.

[Han07] DavidJ. Hand. Principles of data mining. *Drug Safety*, pages 621–622, 2007.

[HK06] J. Han e M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems Series. Elsevier Science & Tech, 2006. URL: http://books.google.at/books?id=AfL0t-YzOrEC.

[KBS05] Dirk Krafzig, Karl Banke e Dirk Slama. *Enterprise SOA: Service Oriented Architecture Best Practices*. Prentice Hall Professional Technical Reference, Upper Saddle River, NJ, 8 edition, 2005.

[Kni] Kayla Knight. Usability Testing With Card Sorting. URL: http://sixrevisions.com/usabilityaccessibility/card-sorting/ [último acesso em 01/07/2014].

[Lar04] Daniel T. Larose. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience, 2004.

[Mac13] Gonçalo Jorge Machado. alive panoramics. Technical report, Faculdade de Engenharia da Universidade do Porto, Junho 2013.

[Men11] Armando B. Mendes. *Metodologias de Data Mining*. Influir, 2011.

[MR10] Oded Maimon e Lior Rokach. *Data Mining and Knowledge Discovery Handbook, 2nd ed*. Springer, 2010.

[Nie12] Jakob Nielsen. Usability 101: Introduction to usability. 2012.

[Nor04] Donald A. Norman. *Emotional Design*. New York: Basic Books, 2004.

[OD08] David L. Olson e Dursun Delen. *Advanced Data Mining Techniques*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[Por12] João Pedro Portela. Multi-touch interaction for interface prototyping. Technical report, Faculdade de Engenharia da Universidade do Porto, Julho 2012.

[PPT08] Mykola Pechenizkiy, Seppo Puuronen e Alexey Tsymbal. Does relevance matter to data mining research?. In Tsau Young Lin, Ying Xie, Anita Wasilewska e Churn-Jung Liau, editors, *Data Mining: Foundations and Practice*, volume 118 of *Studies in Computational Intelligence*, pages 251–275. Springer, 2008. URL: http://dblp.uni-trier.de/db/series/sci/sci118.html#PechenizkiyPT08.

[Rib13] Hugo Ribeiro. Usabilidade acessível: metodologias para a avaliação qualitativa da usabilidade no design para a web. Technical report, Faculdade de Belas Artes da Universidade do Porto, Junho 2013.

[RSP02] Y. Rogers, H. Sharp e J. Preece. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley and Sons Ltd, 2002.

[Sah] Shankar Sahai. Big Data - News, Views and Reviews: Not all data mining packages are created equal (Comparison of 6 major free tools). URL: http://www.infoivy.com/2014/06/not-all-data-mining-packages-are.html [último acesso em 20/06/2014].

[Shn97] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1997.

[Tei09] Tiago Mourão Teixeira. Web collaboration for software engineering. Technical report, Faculdade de Engenharia da Universidade do Porto, Julho 2009.

[WF05] Ian H. Witten e Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Elsevier, Amsterdam, second edition, June 2005. URL: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0120884070.

REFERÊNCIAS

Anexo A

Entrevistas e Questionários

Esta secção contém o guião das entrevistas feita para levantamento de requisitos e os resultado transcritos das mesmas. Para além disso é apresentado o questionário feito e alguma informação estatística relativa às respostas obtidas.

A.1 Guião da Entrevista

A Figura A.2 mostra o guião das entrevistas feitas a três profissionais da área de DM, para o levantamento de requisitos.

Guião de Entrevista: Data Mining Teams

Plataforma Colaborativa para Projetos de Data Mining

Esta entrevista enquadra-se no projeto de dissertação intitulado "Data Mining Teams: Plataforma Colaborativa para Projetos de Data Mining", no âmbito do Mestrado em Engenharia Informática e Computação, da Faculdade de Engenharia da Universidade do Porto.

O principal objetivo é obter a perceção dos profissionais de Data Mining, sobre as principais dificuldades sentidas no processo de data mining e dinâmica entre os membros das equipas de trabalho. Com estes dados pretende construir-se uma plataforma que responda tanto quanto possível aos seus interesses e necessidades, colmatando as dificuldades sentidas, facilitando, por exemplo, a partilha de dados de diferentes formatos, provenientes das várias ferramentas de análise de dados. Este problema foi reportado por uma equipa de profissionais de DM, aquando da participação numa competição de data mining.

Assim, gostaria que respondesse às seguintes questões:

1. Caraterísticas pessoais
   - 1.1. Qual a sua idade?
   - 1.2. Quais são as suas habilitações académicas?
   - 1.3. Qual a função profissional que exerce atualmente?
   - 1.4. Quais foram as atividades profissionais que já exerceu?
   - 1.5. Quanto tempo exerceu cada uma das atividades que referiu?
2. Ferramentas de Data Mining
   - 2.1. Quais são as ferramentas de Data Mining que conhece? Quais usa?
   - 2.2. Quais são as principais caraterísticas comuns nas ferramentas que referiu?
   - 2.3. Quais são as principais caraterísticas que distinguem umas ferramentas das outras?
   - 2.4. Quais são os aspetos que mais valoriza em cada uma das ferramentas que conhece?
   - 2.5. Quais são os aspetos que menos valoriza em cada uma das plataformas que conhece?
   - 2.6. Costuma trabalhar em equipas de projetos de Data Mining?
   - 2.7. Quais as principais dificuldades que sente nesse trabalho?
3. Se o convidassem a construir uma plataforma colaborativa de Data Mining:
   - 3.1. Quais eram as funcionalidades que mais privilegiava?
   - 3.2. Quais eram os aspetos que evidenciava no painel principal?
   - 3.3. Como realçava os aspetos mais relevantes na plataforma?
   - 3.4. Qual a posição que atribuía à informação em função do nível de importância?
   - 3.5. Em que posição colocava os menus principais?

Obrigada pela colaboração

Figura A.1: Guião da Entrevista

A.2 Resultados das Entrevistas

As tabelas que se seguem apresentam a informação recolhida em cada uma das entrevistas.

Categorias	Sujeito 1
Trabalha frequentemente com DM	Sim
Trabalha frequentemente com DM	R, Rapidminer, Weka, Sage, Python, SPSS
Ferramentas com que trabalha ou já trabalhou	R e Rapidminer
Características Comuns	- Interface Gráfica; - Workflow dos processos; - Todas cobrem principais tarefas de DM (classificação, regressão, associação, clustering); - Têm algoritmos de Machine Learning e Data Mining; - Acesso a bases de dados relacionais.
Caraterísticas que mais valoriza das ferramentas	R: -Reprodutibilidade dos resultados; - Muita variedade de algoritmos; Rapidminer: -Workflows.
Caraterísticas que distinguem as ferramentas	Rapidminer e R: -Interface do utilizador; - Eficiência Computacional; - Formato interno de armazenamento dos dados; -R: ferramenta de scripting; -Rapidminer: ferramenta de workflows.
Principais desvantagens	Rapidminer: não gosta da implementação dos algoritmos.
Costuma desenvolver projetos em equipa	Sim
Dificuldades sentidas em projetos de DM em equipa	- Dificuldade na troca de dados; - Acesso a um repositório comum; - Metodologias de DM não estão adequadas, não facilitam o trabalho, em particular, a ligação entre os objetivos do projeto; - Dificuldade de partilha de know-how.
Principais funcionalidades para a plataforma	- Facilidade em partilhar dados; - Gestão de conhecimento: o algoritmo aconselhar com base na experiência o que fazer e o que não fazer relativamente aos dados; - Apresentar as operações que já foram executadas sobre os dados; - Apresentar os icons das ferramentas e permitir arrastar os dados e abrir a ferramenta "automaticamente";
Como poderia ser agrupada a informação?	- Existir um conjunto de webservices, disponibilizados pelas ferramentas; - Fluxo de dados, cada um deles mostra as operações que já foram executadas neles; - Organização por projetos;
Importante que a plataforma permita ligação direta às ferramentas	Sim

Categorias	Sujeito 2
Trabalha frequentemente com DM	Sim
Trabalha frequentemente com DM	R, Knime, Weka e Rapidminer
Ferramentas com que trabalha ou já trabalhou	R e Knime
Características Comuns	-
Caraterísticas que mais valoriza das ferramentas	R: - Open-source: usar sem precisar de licenças, poder contribuir para a ferramenta. Knime: - Open-source; - Interface gráfica para utilizador (não exige conhecimentos de programação)
Caraterísticas que distinguem as ferramentas	-
Principais desvantagens	- Bibliotecas para as quais não há correspondência em todas as ferramentas; - Diferentes implementações para o mesmo algoritmo;
Costuma desenvolver projetos em equipa	Não
Dificuldades sentidas em projetos de DM em equipa	- Fazer análise de dados e implementação de algoritmos sem bases de programação.
Principais funcionalidades para a plataforma	- Ser colaborativa; - Integrar diferentes algoritmos implementados de forma standard;
Como poderia ser agrupada a informação?	- Por dados; - Por algoritmos; - Por processamento de dados; - Por processamento de resultados; - Conhecer resultados de formas diferentes;
Importante que a plataforma permita ligação direta às ferramentas	Sim

Categorias	Sujeito 3
Trabalha frequentemente com DM	Sim
Trabalha frequentemente com DM	Moa, Pentaho e ferramentas Microsoft, R
Ferramentas com que trabalha ou já trabalhou	Moa, Pentaho, R
Características Comuns	Ferramentas para lidar com grandes quantidades de dados.
Caraterísticas que mais valoriza das ferramentas	-
Caraterísticas que distinguem as ferramentas	-
Principais desvantagens	-
Costuma desenvolver projetos em equipa	Sim
Dificuldades sentidas em projetos de DM em equipa	- No meio académico uma das dificuldades é o facto de haver muito multitasking, que não permite uma dedicação tão grande ao acompanhamento dos projetos; - Falta de know-how, relativamente às ferramentas, dos elementos das equipas.
Principais funcionalidades para a plataforma	- Encontrar um formato específico para guardar os resultados das ferramentas de data mining.
Como poderia ser agrupada a informação?	
Importante que a plataforma permita ligação direta às ferramentas	-

A.3 Questionário

As figuras seguintes apresentam as perguntas feitas no questionário administrado para o levantamento de requisitos.

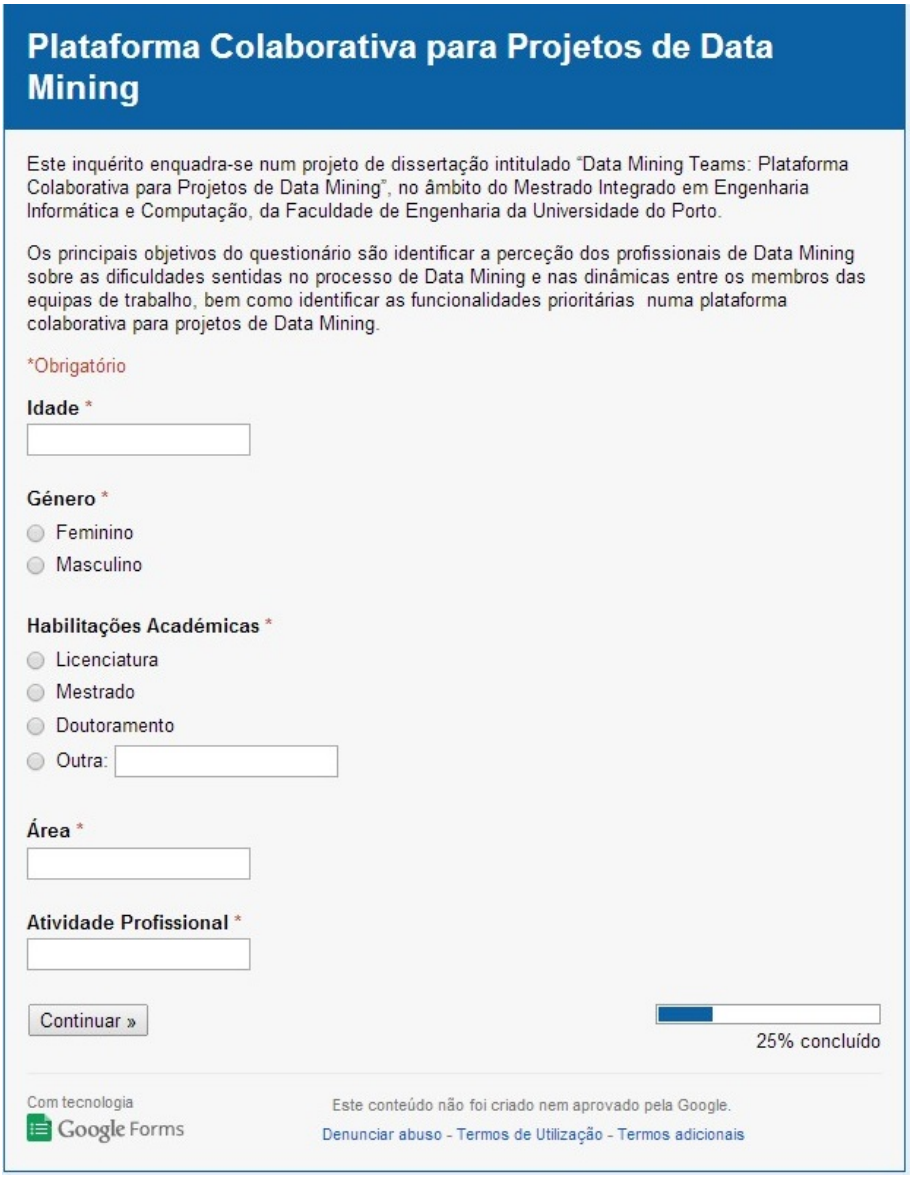


Figura A.2: Questionário Parte 1

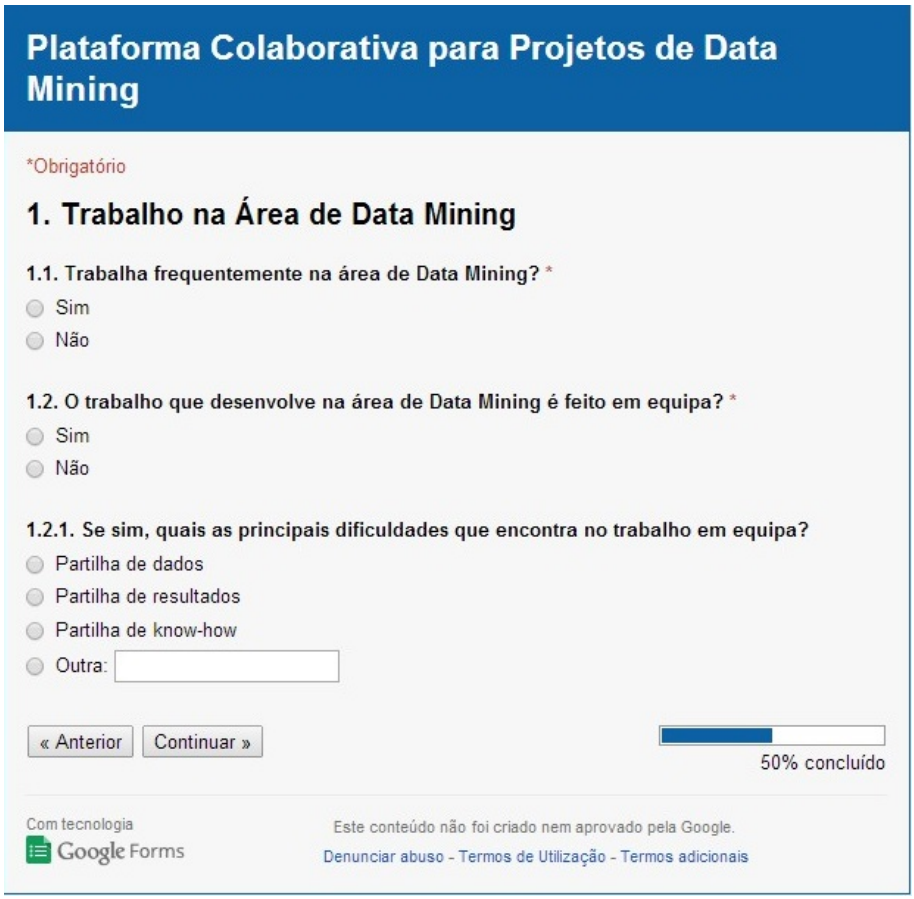
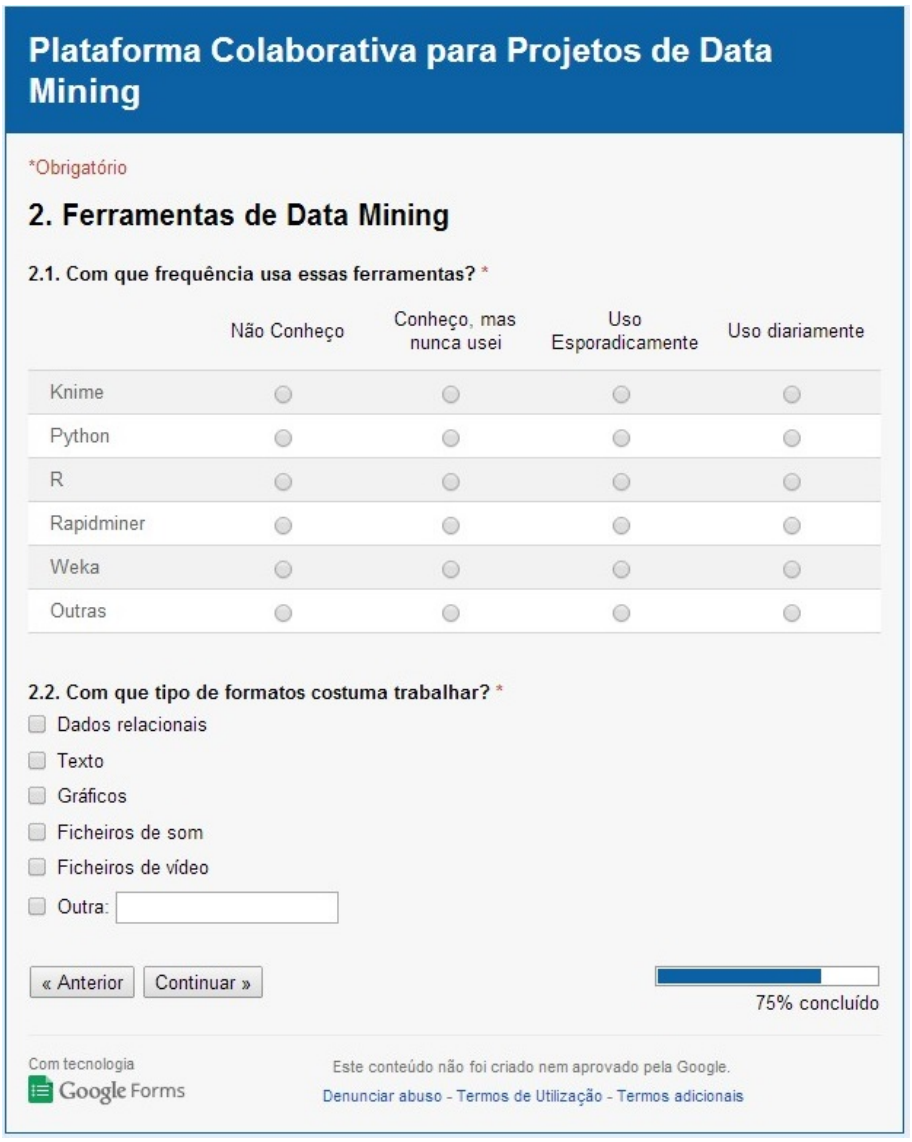


Figura A.3: Questionário Parte 2



Plataforma Colaborativa para Projetos de Data Mining

*Obrigatório

2. Ferramentas de Data Mining

2.1. Com que frequência usa essas ferramentas? *

	Não Conheço	Conheço, mas nunca usei	Uso Esporadicamente	Uso diariamente
Knime	○	○	○	○
Python	○	○	○	○
R	○	○	○	○
Rapidminer	○	○	○	○
Weka	○	○	○	○
Outras	○	○	○	○

2.2. Com que tipo de formatos costuma trabalhar? *

- ☐ Dados relacionais
- ☐ Texto
- ☐ Gráficos
- ☐ Ficheiros de som
- ☐ Ficheiros de vídeo
- ☐ Outra:

« Anterior | Continuar »

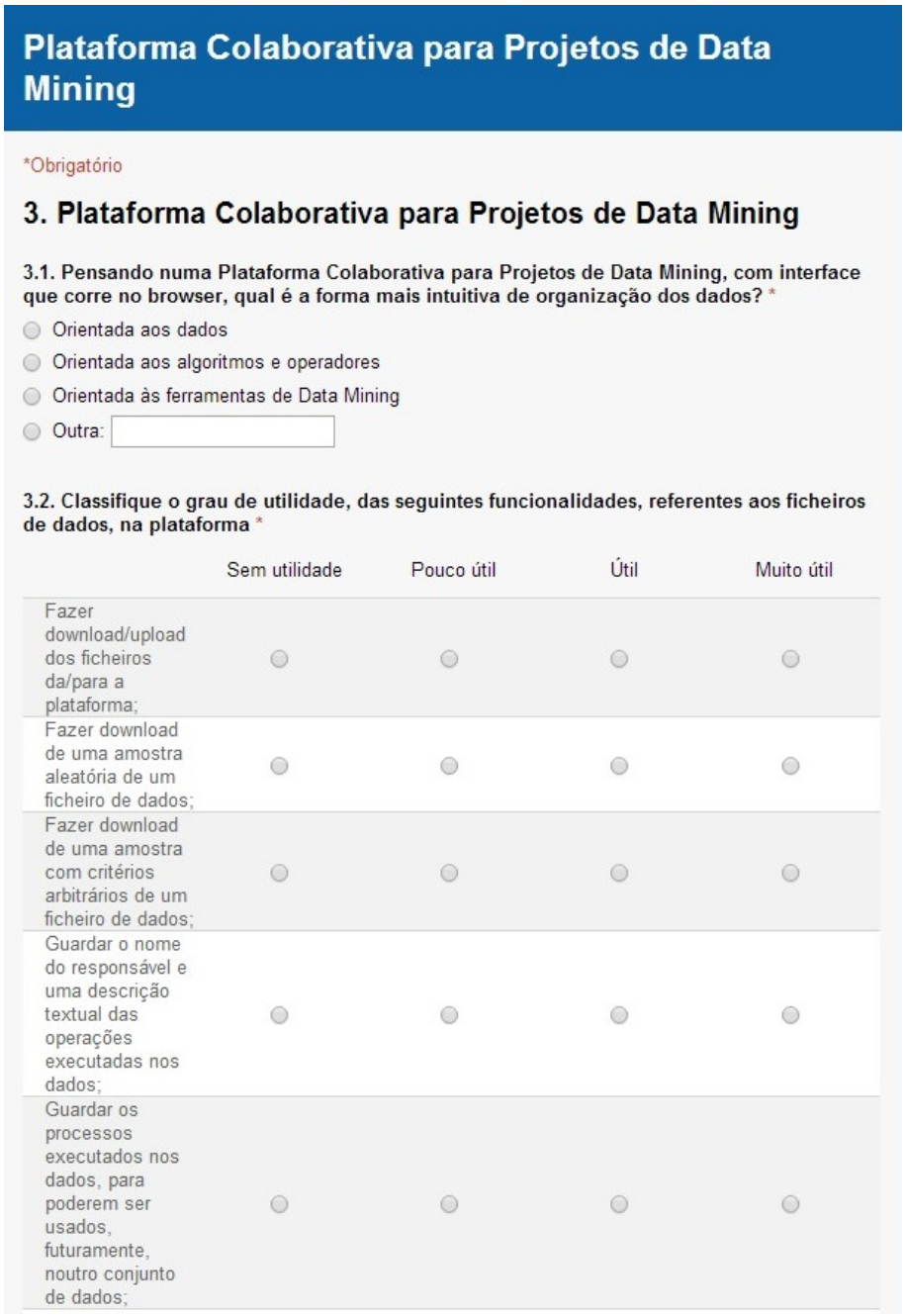
75% concluído

Com tecnologia Google Forms

Este conteúdo não foi criado nem aprovado pela Google.

Denunciar abuso - Termos de Utilização - Termos adicionais

Figura A.4: Questionário Parte 3



Plataforma Colaborativa para Projetos de Data Mining

*Obrigatório

3. Plataforma Colaborativa para Projetos de Data Mining

3.1. Pensando numa Plataforma Colaborativa para Projetos de Data Mining, com interface que corre no browser, qual é a forma mais intuitiva de organização dos dados? *

- ○ Orientada aos dados
- ○ Orientada aos algoritmos e operadores
- ○ Orientada às ferramentas de Data Mining
- ○ Outra: _____

3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma *

	Sem utilidade	Pouco útil	Útil	Muito útil
Fazer download/upload dos ficheiros da/para a plataforma;	○	○	○	○
Fazer download de uma amostra aleatória de um ficheiro de dados;	○	○	○	○
Fazer download de uma amostra com critérios arbitrários de um ficheiro de dados;	○	○	○	○
Guardar o nome do responsável e uma descrição textual das operações executadas nos dados;	○	○	○	○
Guardar os processos executados nos dados, para poderem ser usados, futuramente, noutro conjunto de dados;	○	○	○	○

Figura A.5: Questionário Parte 4

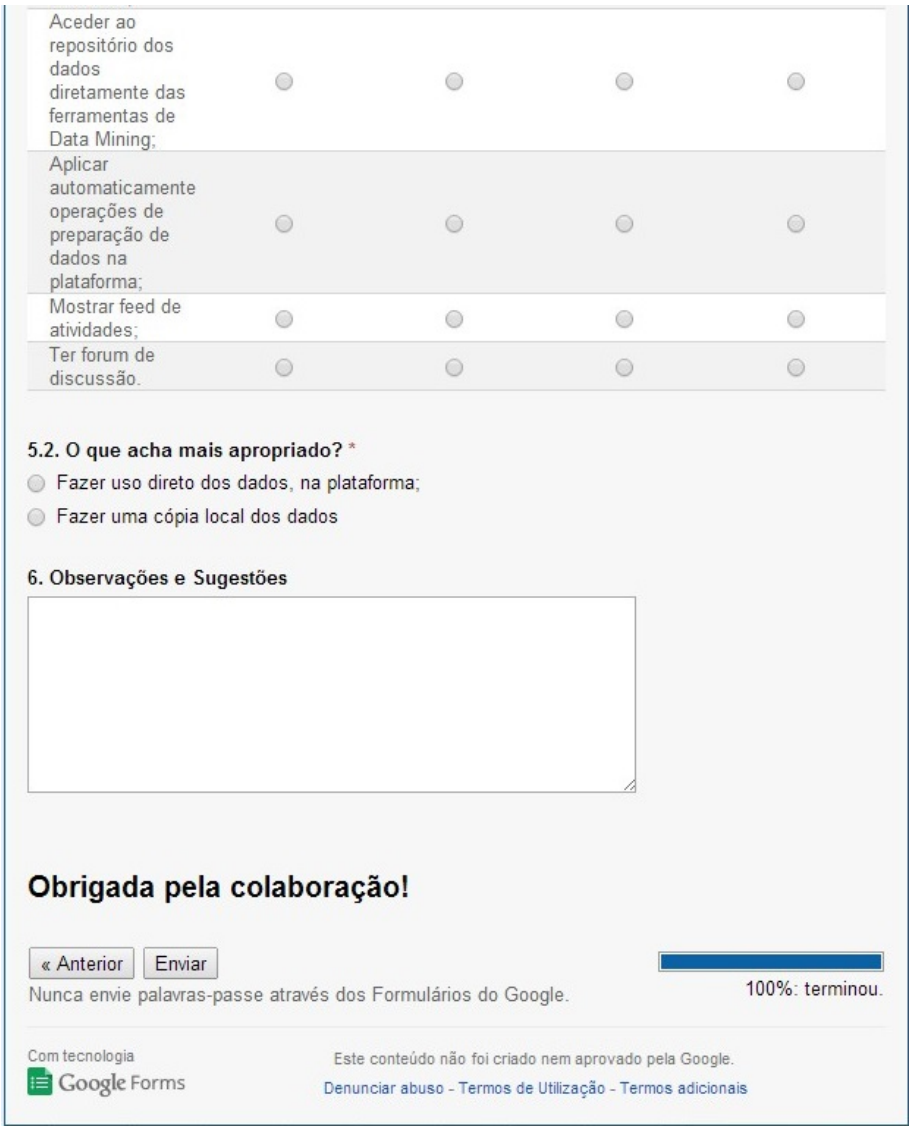
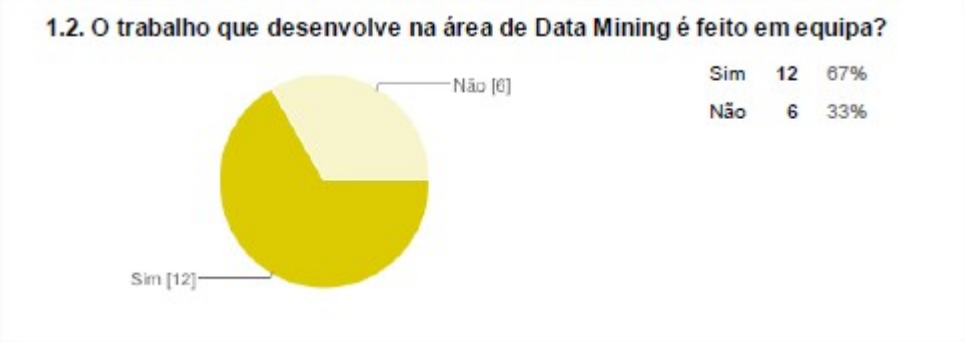


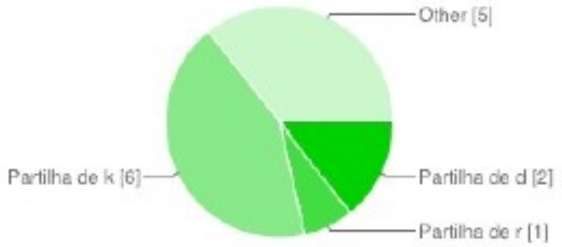
Figura A.6: Questionário Parte 4 (continuação)

A.4 Resultados dos Questionários

De seguida são apresentados os gráficos com a informação relativa à informação recolhida através da administração dos questionários.



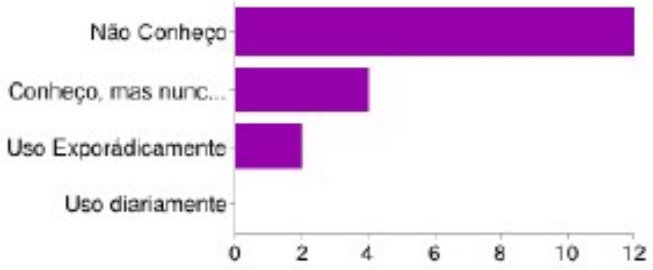
1.2.1. Se sim, quais as principais dificuldades que encontra no trabalho em equipa?



Partilha de dados	2	14%
Partilha de resultados	1	7%
Partilha de know-how	6	43%
Other	5	36%

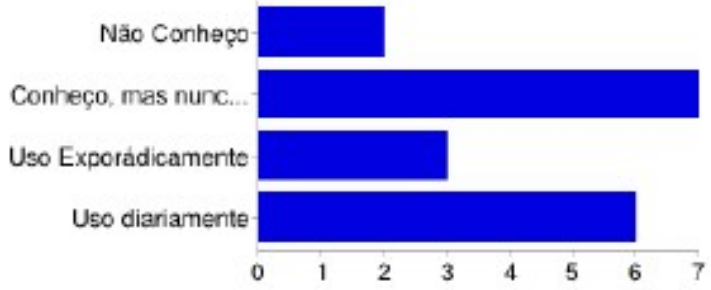
2. Ferramentas de Data Mining

Knime [2.1. Com que frequência usa essas ferramentas?]



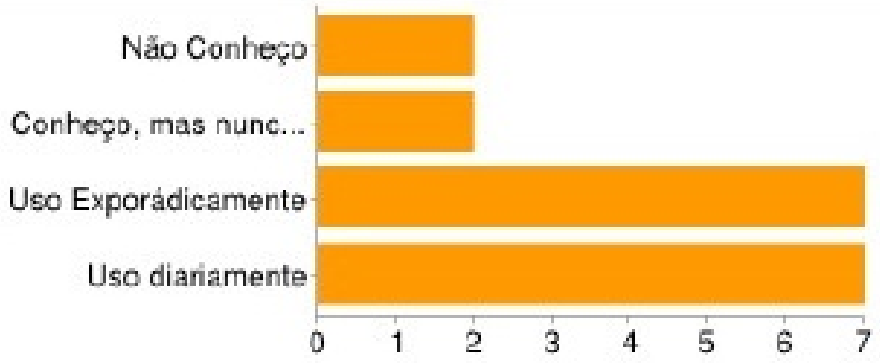
Não Conheço	12	67%
Conheço, mas nunca usei	4	22%
Uso Exporádicamente	2	11%
Uso diariamente	0	0%

Python [2.1. Com que frequência usa essas ferramentas?]



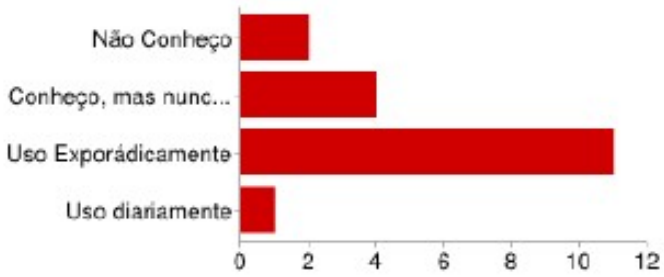
Não Conheço	2	11%
Conheço, mas nunca usei	7	39%
Uso Exporádicamente	3	17%
Uso diariamente	6	33%

R [2.1. Com que frequência usa essas ferramentas?]



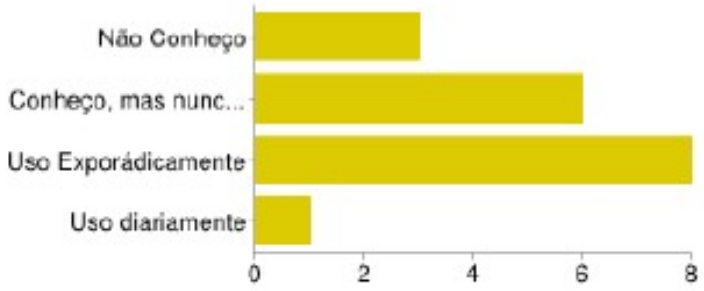
Não Conheço	2	11%
Conheço, mas nunca usei	2	11%
Uso Exporádicamente	7	39%
Uso diariamente	7	39%

Rapidminer [2.1. Com que frequência usa essas ferramentas?]



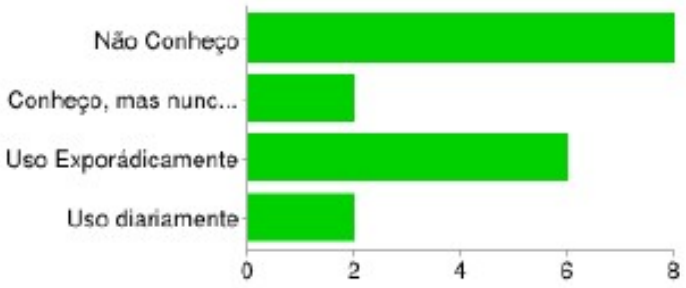
Não Conheço	2	11%
Conheço, mas nunca usei	4	22%
Uso Exporádicamente	11	61%
Uso diariamente	1	6%

Weka [2.1. Com que frequência usa essas ferramentas?]



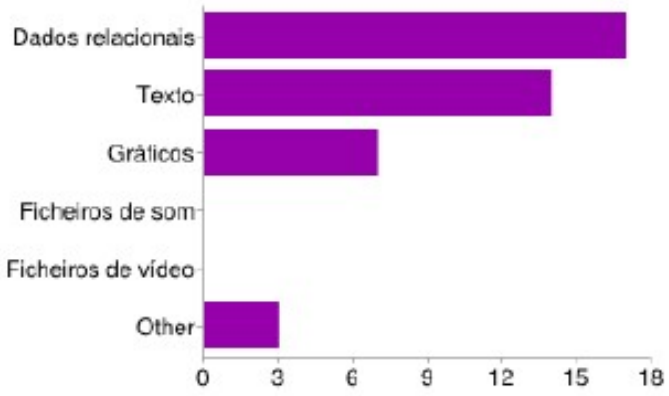
Não Conheço	3	17%
Conheço, mas nunca usei	6	33%
Uso Exporádicamente	8	44%
Uso diariamente	1	6%

Outras [2.1. Com que frequência usa essas ferramentas?]



Não Conheço	8	44%
Conheço, mas nunca usei	2	11%
Uso Exporádicamente	6	33%
Uso diariamente	2	11%

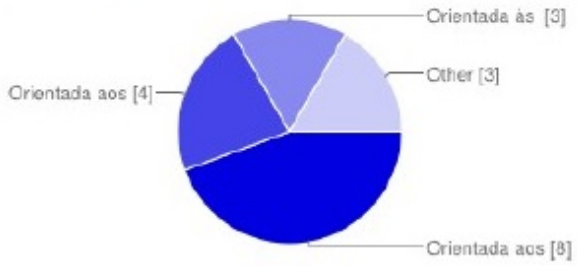
2.2. Com que tipo de formatos costuma trabalhar?



Dados relacionais	**17**	41%
Texto	**14**	34%
Gráficos	**7**	17%
Ficheiros de som	**0**	0%
Ficheiros de vídeo	**0**	0%
Other	**3**	7%

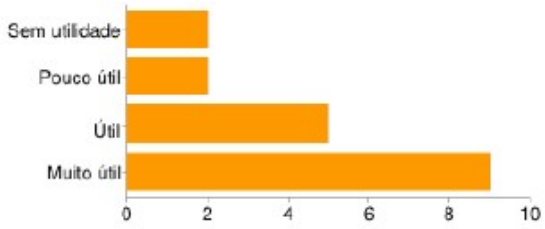
3. Plataforma Colaborativa para Projetos de Data Mining

3.1. Pensando numa Plataforma Colaborativa para Projetos de Data Mining, com interface que corre no browser, qual é a forma mais intuitiva de organização dos dados?



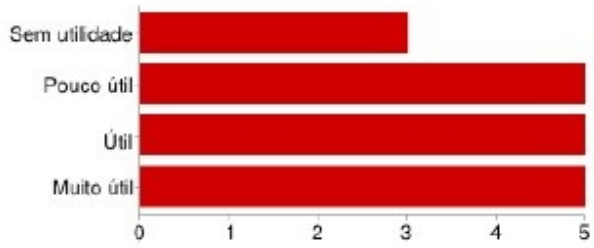
Orientada aos dados	8	44%
Orientada aos algoritmos e operadores	4	22%
Orientada às ferramentas de Data Mining	3	17%
Other	3	17%

Fazer download/upload dos ficheiros da/para a plataforma; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



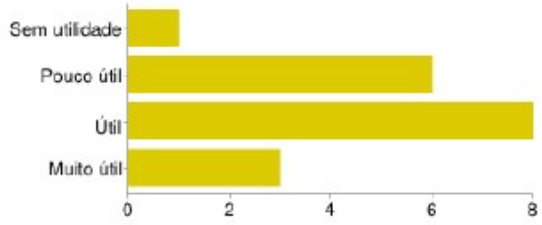
Sem utilidade	2	11%
Pouco útil	2	11%
Útil	5	28%
Muito útil	9	50%

Fazer download de uma amostra aleatória de um ficheiro de dados; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



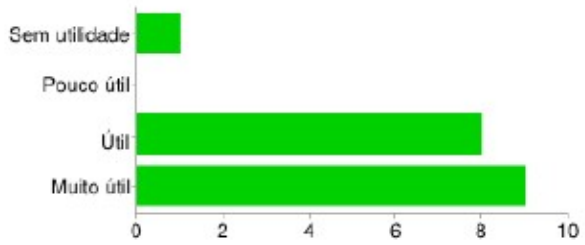
Sem utilidade	3	17%
Pouco útil	5	28%
Útil	5	28%
Muito útil	5	28%

Fazer download de uma amostra com critérios arbitrários de um ficheiro de dados; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



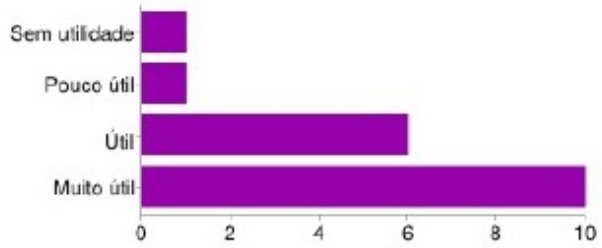
Sem utilidade	1	6%
Pouco útil	6	33%
Útil	8	44%
Muito útil	3	17%

Guardar o nome do responsável e uma descrição textual das operações executadas nos dados; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



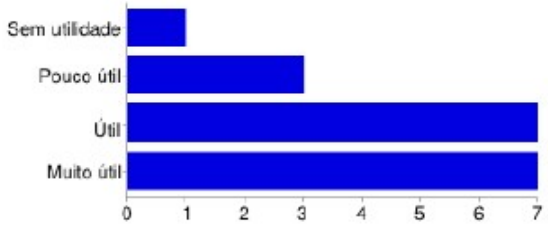
Sem utilidade	1	6%
Pouco útil	0	0%
Útil	8	44%
Muito útil	9	50%

Guardar os processos executados nos dados, para poderem ser usados, futuramente, noutro conjunto de dados; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



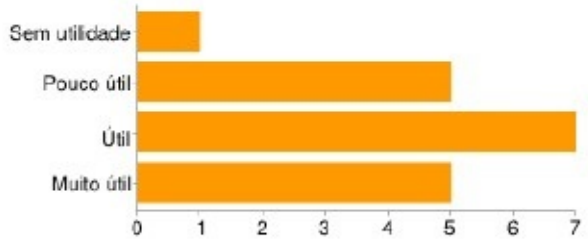
Sem utilidade	1	6%
Pouco útil	1	6%
Útil	6	33%
Muito útil	10	56%

Aceder ao repositório dos dados diretamente das ferramentas de Data Mining; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



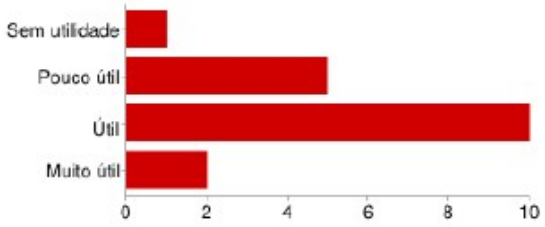
Sem utilidade	1	6%
Pouco útil	3	17%
Útil	7	39%
Muito útil	7	39%

Aplicar automaticamente operações de preparação de dados na plataforma; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



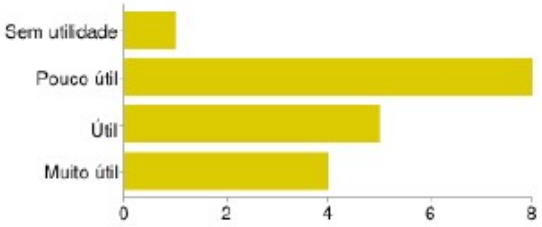
Sem utilidade	1	6%
Pouco útil	5	28%
Útil	7	39%
Muito útil	5	28%

Mostrar feed de atividades; [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



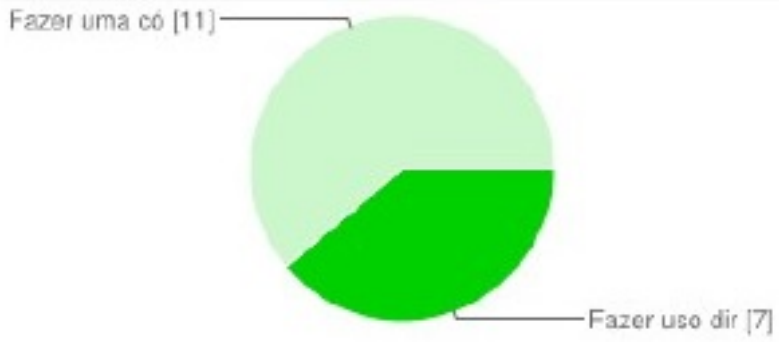
Sem utilidade	1	6%
Pouco útil	5	28%
Útil	10	56%
Muito útil	2	11%

Ter forum de discussão. [3.2. Classifique o grau de utilidade, das seguintes funcionalidades, referentes aos ficheiros de dados, na plataforma]



Sem utilidade	1	6%
Pouco útil	8	44%
Útil	5	28%
Muito útil	4	22%

5.2. O que acha mais apropriado?



Fazer uso direto dos dados, na plataforma;	**7**	39%
Fazer uma cópia local dos dados	**11**	61%