INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Ferramenta de Data Mining para Dados Educacionais

DIOGO MANUEL PEREIRA VIEIRA
Outubro de 2018

POLITÉCNICO
DO PORTO

Ferramenta de Data Mining para Dados Educacionais

Diogo Manuel Pereira Vieira

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas Computacionais

Porto, Outubro 2018

Ferramenta de Data Mining para Dados Educacionais

Diogo Manuel Pereira Vieira

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas Computacionais

Orientador: Elsa Ferreira Gomes

Coorientadora: Paula Tavares

Agradecimentos

Uma vez concluído este trabalho, não queria deixar de mencionar aqueles que contribuíram ao longo do projeto e que me ajudaram a atingir este objetivo.

A professora Elsa Ferreira Gomes e coorientadora Paula Tavares pela ajuda, conselhos, disponibilidade e dedicação ao longo da dissertação.

Aos meus pais pelo apoio incansável, por toda a força e sacrifício que fizeram para eu terminar este projeto.

À minha namorada Joana pela paciência e insistência em finalizar o projeto, assim como todo o carinho e ajuda em todo o processo.

A todos os meus colegas que fizeram parte do percurso, desde o início da licenciatura até ao mestrado, pela companhia, apoio e compreensão.

Resumo

Data mining representa o processo desenvolvido para examinar grandes quantidades de dados e também se refere a uma coleção de ferramentas usadas para executar o seu processo.

As regras de associação são uma tarefa importante no processo de *data mining* que consiste em retirar informação sobre as relações presentes nos dados analisados. Existem vários algoritmos para o tratamento e exploração das regras de associação que, por norma, geram um número de regras substancialmente grande, tornando assim o processo de pós-processamento das regras uma tarefa mais complicada e morosa.

Na presente dissertação foi efetuado um estudo sobre o processo de *data mining* na educação, o processo de extração de regras de associação e os seus algoritmos e, no final, foi desenvolvida uma ferramenta que permite gerar e explorar regras de associação a partir de um conjunto de dados.

A ferramenta desenvolvida, denominada Ferramenta de Exploração de Regras de Associação (**FERA**)**,** tem como objetivo auxiliar o utilizador a gerar regras mais interessantes, visualizando as regras geradas através de gráficos e navegando por essas mesmas regras. A ferramenta é facilmente extensível, dispondo de vários tipos de visualização, dando a possibilidade ao utilizador de guardar a informação gerada.

Na presente dissertação é efetuado uma avaliação da ferramenta desenvolvida a dados educacionais, de forma a demonstrar a utilidade desta e o seu propósito.

Palavras-chave: *data mining*, FERA, RStudio, regras de associação.

Abstract

Data mining represents the process designed to examine large amounts of data and refers to a collection of tools used to run your process. The Association Rules are an important task in the data mining process, which consists of extracting information about the relationships present in the analyzed data. There are several algorithms for handling and exploiting Association Rules which, as a rule, generate a substantially large number of rules, thus making the process of post-processing rules a more complicated and time-consuming task.

In this dissertation, a study was carried out on the data mining process in education, the process of extracting Association Rules and their algorithms and, in the end, a tool was developed that allows generating and exploring Association Rules from a set of data.

The developed tool, called **FERA**, aims to help the user to generate more interesting rules, visualizing the rules generated by graphs and navigating the same rules. The tool is easily extensible, having several types of visualization, allowing the user to save the generated information.

In this dissertation is made an evaluation of the application on educational data, in order to demonstrate the benefit of it and its purpose.

Keywords: data mining, FERA, RStudio, association rules.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

XML	*Extensible Markup Language*
CSV	*Comma separated values*
FERA	Ferramenta Exploração de Regras de Associação
ISEP	Instituto Superior de Engenharia do Porto
APROG	Algoritmia e Programação
EDM	*Educational Data Mining*
HTML	*Hypertext Markup Language*
PDF	*Portable Document Format*
PEAR	*Post-Processing Environment for Association Rules*
NCD	*New Concept Development Model*
KDD	*Knowledge Discovery in Databases*
SSL	*Security Socket Layer*
HTTPS	*Hypertext Transfer Protocol Secure*
HTTP	*HyperText Transfer Protocol*

1 Introdução

Neste capítulo, são contextualizados a motivação do projeto, é apresentado, de forma detalhada, o problema proposto para a realização do projeto, os objetivos, a abordagem que foi tomada e é descrita a estrutura que o presente documento contém.

1.1 Motivação

Atualmente existem muitos alunos a frequentar o Ensino Superior na área de Engenharia Informática que cada instituição oferece. Porém, muitos alunos perdem o interesse após iniciarem o ensino, demorando mais tempo a finalizar o seu curso ou acabando por desistir (Engrácia & Baptista, 2018). Por estes motivos, considerou-se desenvolver uma ferramenta de *data mining* para extração e visualização de regras de associação, que permite analisar dados provenientes de ficheiros XML/CSV com informação de inquéritos realizados a alunos.

1.2 Apresentação do Problema

O projeto em questão, consiste na criação de uma ferramenta de *data mining*, permitindo a análise a resultados de inquéritos a estudantes disponibilizando gráficos e deteção de *outliers* usando técnicas de *data mining* como as regras de associação e métodos de previsão.

Em particular, esta ferramenta permite analisar dados de inquéritos realizados a alunos, provenientes de ficheiros XML/CSV, tendo como propósito extrair conhecimento sobre os dados analisados no sentido de implementar mudanças na estratégia abordada atualmente.

Assim, nesta dissertação apresenta-se uma **F**erramenta para **E**xploração de **R**egras de **A**ssociação (**FERA**) para geração, visualização e extração de regras de associação, em que o utilizador consegue gerar regras de acordo com os parâmetros selecionados e de acordo com o seu conhecimento prévio sobre os dados inseridos.

A vantagem de existir uma ferramenta como a **FERA** consiste na facilidade de visualização das regras de associação extraídas e na escolha ou geração de regras mais interessantes para o caso de estudo, de acordo com os filtros selecionados na ferramenta. Contém também a facilidade de extrair todas as regras geradas e os gráficos gerados para um ficheiro *hypertext markup language* (HTML).

1.3 Objetivo

O objetivo do projeto que aqui se descreve consiste na realização de uma ferramenta (a FERA) para auxiliar o utilizador a gerar regras de associação interessantes, de acordo com o seu objetivo. Para tal, permite visualizar as regras de associação geradas através de gráficos que fornecem informação relevante sobre essas regras.

Em particular, permite que o utilizador analisar dados educacionais provenientes de ficheiros XML/CSV e retirar conclusões que auxiliem a prever o comportamento dos estudantes e melhore as técnicas de aprendizagem dos estudantes, em particular, na aprendizagem de informática.

1.4 Abordagem

A solução do problema passa pelo desenvolvimento de uma ferramenta de análise de dados, desenvolvida em R, que permita retirar conclusões dos dados analisados.

Para realizar o processo de avaliação foram definidas algumas grandezas que irão ajudar a avaliar a solução aqui documentada, sendo elas:

- **Inquéritos de satisfação**: utilizada para recolher os dados a serem analisados pela ferramenta a desenvolver;
- **Desempenho**: utilizado para medir o desempenho da máquina ao processar a ferramenta;
- **Conhecimento do utilizador**: através do conhecimento prévio do utilizador, é possível identificar se o conhecimento adquirido após a utilização da ferramenta é interessante ou não.

Os dados a serem analisados refletem o grau de satisfação dos estudantes em relação aos métodos de aprendizagem, serviços que uma instituição disponibiliza para um bom desempenho dos estudantes no seu curso e também o desempenho dos docentes de ensino no seu acompanhamento e métodos de ensino.

1.5 Estrutura

O presente documento contém uma estrutura organizada onde o presente capítulo pretende fazer uma descrição do projeto a desenvolver e fornecer uma contextualização genérica do problema e da solução da proposta.

No capítulo Análise de Negócio está contida a análise de valor da ferramenta desenvolvida. Este capítulo contém o processo de negócio, Modelo Canvas e é seguido pelo capítulo Estado da Arte, que contém uma análise sobre alguns estudos de aplicação de técnicas de *data mining* a dados educacionais e uma análise sobre uma ferramenta de extração de regras de associação, semelhante à **FERA**.

No capítulo seguinte, Extração de conhecimento de dados, é efetuada uma contextualização dos conceitos aprendidos e estudados durante o projeto, que contém os subcapítulos Data Mining, Regras de Associação, Algoritmos de extração de regras de associação e Pós-processamento de regras de associação.

No capítulo Tecnologias utilizadas é feita uma descrição das tecnologias utilizadas ao longo do desenvolvimento do projeto.

No capítulo Design da solução é desenhada a solução implementada, tendo como base os conceitos definidos no capítulo 4 e os requisitos da ferramenta.

O capítulo FERA retrata a solução implementada assim como os exemplos de aplicação que se efetuaram e os resultados obtidos.

O último capítulo do documento, Conclusões, são identificadas as conclusões da ferramenta desenvolvida, as dificuldades ultrapassadas, as limitações encontradas, o trabalho futuro e a apreciação global do projeto.

2 Análise de Negócio

Este capítulo tem como principal objetivo identificar os elementos chaves do negócio, o modelo Canvas e uma análise de abordagens já existentes similares ao projeto em questão.

2.1 Processo de Negócio

Para realizar o processo de negócio deste projeto foi aplicado o modelo NCD (New Concept Development Model). Este modelo foi desenvolvido por Peter Koen (Koen, 2004) e é usado para descrever as etapas de análise, criação e inovação de um produto. O modelo NCD (Figura 1) contém cinco elementos-chave que serão descritas abaixo de acordo com o âmbito deste projeto.

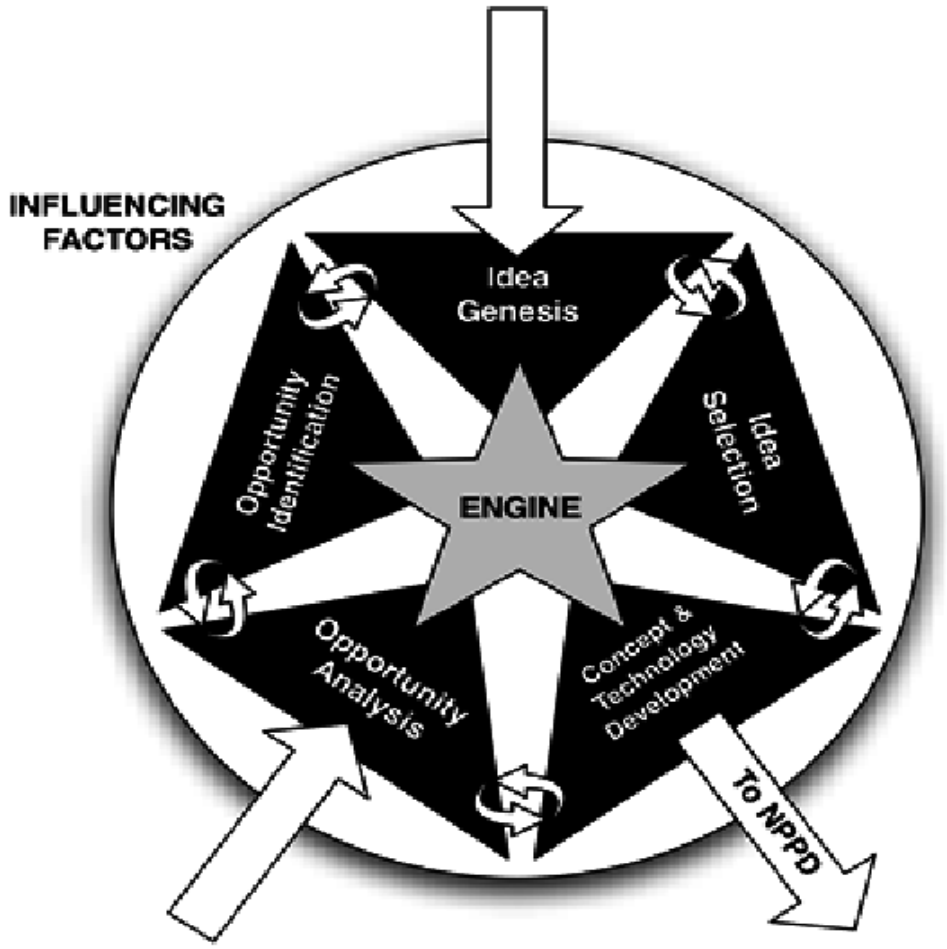

Figura 1 - Modelo New Concept Development Model (Industrial Research Institute, Inc, 2001)

1. **Identificação de Oportunidade**

Todas as instituições de ensino lidam com a mudança de curso/instituição dos seus alunos e também com a retenção dos seus alunos nos vários cursos lecionados.

Pelo facto de haver mais alunos a entrarem no ramo da informática, sentiu-se a necessidade de desenvolver uma ferramenta que analise dados de inquéritos de satisfação de alunos e extrair conhecimento, de forma a perceber os erros no ensino e adotar novas estratégias ou aperfeiçoar as existentes, de modo a melhorar aprendizagem dos alunos.

2. **Análise de Oportunidade**

Apesar de já existirem ferramentas de casos de estudo na área de *data mining* e uma ferramenta denominada por *Post-Processing Environment for Association Rules* (PEAR), para o pós-processamento de regras de associação, continua a existir a oportunidade de criar uma ferramenta com menos limitações que a referida anteriormente e com outras funcionalidades.

O que distingue este negócio dos outros é o facto de a ferramenta conseguir processar dados de inquéritos em ficheiros XML/CSV e ser vocacionado a um nível educacional, isto é, os dados a serem analisados podem ser a nível de institucional para assim identificar, numa determinada instituição, os métodos a aplicar para melhorar os pontos referidos acima.

3. **Geração de Ideias**

O negócio engloba-se na área de *Educational Data Mining* (EDM) que tem como objetivo aplicar técnicas e métodos de *data mining* para melhorar questões relacionadas com a educação, através de algoritmos de regras de associação.

4. **Seleção de Ideias**

Tendo em conta que a aplicação necessita de prever resultados futuros, torna-se imperativo o uso de técnicas de d*ata mining*. A ferramenta desenvolvida irá analisar os dados que foram submetidos para análise, no qual é aplicado o algoritmo Apriori (Cap. 4.3.3), sendo imperativo na ferramenta a visualização das regras de associação geradas.

5. **Definição de Conceito**

A ferramenta baseia-se numa ferramenta *web* capaz de extrair regras de associação de ficheiros de dados educacionais e ser capaz de apresentar as regras de associação ao utilizador. Com a ferramenta, é possível aplicar filtros para serem assim geradas regras mais interessantes para o utilizador, identificando problemas no caso em estudo. Por exemplo, se analisarmos dados

relacionados com o desempenho dos estudantes na aprendizagem e escolhermos filtros com regras que contenham o atributo 'motivação', podemos assim chegar à conclusão, baseado na extração de regras de associação e no conhecimento adquirido por elas, que um aluno motivado aprende mais facilmente os conhecimentos que são lecionados.

Após o conhecimento adquirido pelas regras de associação, é possível resolver problemas na reprovação dos estudantes e na mudança de curso/instituição pela parte dos estudantes, auxiliando as instituições e os seus docentes a melhorar os métodos de ensino, adotando e/ou melhorando os seus métodos de ensino já existentes (Merceron & Yacef, 2005).

2.2 Modelo Canvas

Com a ferramenta desenvolvida pretende-se criar valor para as várias instituições de Ensino Superior, através do conhecimento adquirido da análise da ferramenta, adotando e melhorando os seus métodos de aprendizagem, criando valor para os vários docentes que orientam os alunos na aprendizagem e para as instituições de Ensino Superior que conseguem eventualmente reter mais alunos.

Para as instituições e para os seus docentes é pretendido, com o uso desta ferramenta, subir a taxa de retenção dos alunos na instituição e melhorar a taxa de aprovação dos alunos nas várias disciplinas. Se as instituições implementarem novos métodos de ensino baseados no conhecimento que advém da **FERA,** então é também criado valor para os alunos.

Como os novos métodos de ensino são planeados para um ensino mais aliciante, os alunos conseguirão interiorizar melhor os conhecimentos lecionados.

No caso das instituições que adotarem a ferramenta **FERA**, se o cliente não conseguir utilizar a ferramenta por si só, a relação entre a instituição e os autores da **FERA,** será proactiva, isto é, os inquéritos pelo qual pretende-se fazer para extrair os dados, serão desenvolvidos de acordo com as características da instituição, dos seus docentes de ensino, dos estudantes e também dos serviços que a instituição fornece. Como cada instituição contém as suas particularidades, é preciso ter uma relação proactiva com as mesmas, no sentido de melhorar os seus métodos de ensino junto de cada cliente/instituição e explorar os pontos fracos de cada um.

Com o desenvolvimento da ferramenta, as únicas receitas serão provenientes da compra da ferramenta, caso a instituição queira se tornar independente, ou então dos serviços que são

precisos para a análise correta da ferramenta, isto é, a construção dos inquéritos associados às necessidades de cada cliente, a análise dos dados dos inquéritos realizados e a construção de novos métodos no ensino da instituição.

A componente essencial para criar valor serão os dados, sem a criação ou análise de dados a ferramenta deixa de conseguir criar valor para os vários segmentos. Logo as atividades principais serão a criação e os resultados dos inquéritos, a análise dos dados, definindo os filtros adequados, e os resultados da análise.

O custo do desenvolvimento da ferramenta e dos inquéritos é nulo, isto porque os inquéritos poderão ser realizados com o *google forms* e extraídos em ficheiros XML/CSV, o que reduz o custo a zero na impressão dos inquéritos. A ferramenta de análise é construída através da ferramenta RStudio, ferramenta de *software* que também ela é disponibilizada gratuitamente. Os únicos custos advêm da compra do servidor para alocar a ferramenta *web,* na construção e na divulgação desta.

No final, a ferramenta desenvolvida é o produto. O seu foco passa por melhorar o ensino e os métodos de ensino, não só para um cliente, mas sim a nível global. É importante realçar que o produto desenvolvido pode também ser adaptado para outras áreas de negócio, quer seja a melhorar métodos de trabalho ou melhorar volume de vendas.

3 Estado da Arte

Este capítulo visa identificar alguns artigos de *data mining* e a sua aplicação a dados educacionais, analisando com mais detalhe os artigos presentes nas subseções deste capítulo.

Dois benefícios da exploração de dados aplicados à educação é prever a taxa de sucesso dos alunos e os fatores que os leva a optarem pela transferência para outras instituições de ensino superior ou ao abandono.

Encontram-se vários trabalhos de autores que descrevem a aplicação de *data mining* (assunto tratado no capítulo 4) a dados de estudantes e, em particular, que dizem respeito à motivação. É o caso de Kularbphettong e Tongsiri (Kularbphettong & Tongsiri, 2012) que usam as regras de associação e de classificação para prever o comportamento dos estudantes em várias disciplinas.

Também Gomes et.al (Gomes, et al., 2018) aplicaram regras de associação (secção 4.2) para analisar dados resultantes de inquéritos aos estudantes para extrair regras que permitissem compreender a relação entre a motivação e as dificuldades na aprendizagem de disciplinas de programação ou algoritmos.

No artigo de Amjad Abu Saa (Saa, 2016) é efetuado um estudo para descobrir relações entre os fatores pessoais/sociais dos estudantes com a sua performance no semestre anterior. No estudo é utilizado inquéritos com questões académicas, sociais e pessoais que mais tarde são analisados por processos de *data mining*, nomeadamente técnicas de classificação, para prever as notas dos alunos no final do semestre.

Nos trabalhos apresentados pelos autores Nikolovski (Vlatko Nikolovski, 2014) e por Merceron e Yacef (Merceron & Yacef, 2005) foram desenvolvidas e testadas técnicas de *data mining* a dados educacionais. Em ambos os artigos as técnicas de *data mining* foram aplicadas de forma a extrair conhecimento sobre a qualidade da aprendizagem, tendo como objetivo aumentar a taxa de sucesso na previsão dos parâmetros alvo, ajudando assim os alunos a adquirir conhecimento de um modo mais facilitado (Merceron & Yacef, 2005), auxiliar as instituições de ensino na compreensão do progresso dos alunos e auxiliar as instituições de ensino superior a reter os alunos nos seus cursos (Vlatko Nikolovski, 2014).

De seguida são analisados dois casos de estudos relacionados com *educational data mining*, seguido de um caso de estudo de uma ferramenta de pós-processamento de regras de associação, denominado PEAR, *Post-Processing Environment for Association Rules* (Rocha, 2006).

3.1 Educational Data Mining: A Case Study

O primeiro caso de estudo analisado consiste em aplicar algoritmos de *data mining* para ajudar a descobrir conhecimentos pedagogicamente relevantes contidos em bases de dados de sistemas educacionais (Merceron & Yacef, 2005). Os conhecimentos obtidos são usados para auxiliar os docentes nos seus métodos de ensino, entendendo melhor a forma de como os alunos aprendem e refletindo sobre os seus métodos de ensino fornecendo *feedback* proativo aos alunos.

O objetivo do caso de estudo é sintetizar e compartilhar as experiências dos autores na área para apoiar a reflexão sobre o ensino e a aprendizagem, contribuindo assim para o surgimento de direções estereotipadas.

Os autores do artigo em questão usaram a ferramenta **Clementine** (Pujari & Gupta, 2012), ferramenta de *data mining* que visa permitir os usuários a executar a sua própria mineração de dados que contém uma programação visual e uma *interface* de fluxo de dados que simplifica o processo de *data mining*. Para fazer o agrupamento de dados utilizaram a plataforma **Tada-ED** (Agathe & Yacef, 2004), plataforma dedicada aos professores permitindo-lhes visualizar e explorar trabalhos com o objetivo de descobrir padrões pedagogicamente relevantes, para classificação das regras de associação.

Os algoritmos de agrupamento (*cluster*) têm como propósito encontrar grupos homogéneos de dados. Foi usado o método de agrupamento *k-clustering* (Sayad, 2006) que é um método de quantização vetorial e que visa dividir N observações em K grupos. Foi também utilizado o algoritmo de agrupamento hierárquico (Sayad, 2006) que visa construir uma hierarquia nos diferentes grupos (clusters).

Os algoritmos de classificação, no artigo, são usados para prever os valores das variáveis, como por exemplo, se um determinado aluno realizou todo o trabalho na disciplina, então pode-se prever que o aluno terá um bom desempenho no exame final. Por fim, foram utilizadas regras de associação para encontrar relações entre as variáveis.

O artigo conclui que a descoberta de diferentes padrões através de diferentes algoritmos de *data mining* e técnicas de visualização sugere uma política pedagógica simples. A exploração de dados focada no número de tentativas de exercícios combinados com a classificação levou a identificar os estudantes em risco. Os algoritmos de agrupamento e visualização de *clusters* levaram a identificar um comportamento particular entre os estudantes que reprovam.

3.2 Case Study for Predicting Students dropout

Este caso de estudo tenta encontrar padrões dos dados analisados para prever a desistência dos alunos nas instituições de ensino superior (Vlatko Nikolovski, 2014). O referido caso de estudo analisa dados de uma instituição em particular, relativos a 3 anos consecutivos, que contém detalhes sobres os estudantes, a retenção no curso e a evolução das suas notas nas várias disciplinas. No artigo são aplicados algoritmos de *data mining* e, no final, é ilustrado um modelo capaz de prever os subconjuntos de alunos com tendência a abandonar os estudos após o seu primeiro ano.

Os algoritmos de *data mining* utilizados foram os algoritmos de classificação Naïve Bayes e J48 (Kamber, 2006). O algoritmo de Naïve Bayes tem como objetivo prever probabilidades de associações de classes, isto é, prever a probabilidade que um determinado conjunto de dados contém para uma classe em particular.

O classificador Naïve Bayes analisa todos os atributos contidos individualmente como se fossem igualmente importantes e independentes um do outro. No processo de classificação, cada atributo funciona independentemente dos outros atributos contidos no modelo. Além do tratamento independente dos atributos, o classificador J48 é um modelo preditivo que prevê um atributo como uma variável dependente dos valores de todos os outros atributos. Para classificar um novo item, o algoritmo J48 cria uma árvore de decisão com base nos atributos para ganhar equilíbrio, flexibilidade e precisão.

O caso de estudo mostra que a precisão de ambos os algoritmos dependem, em particular, da qualidade dos atributos extraídos dos dados. A precisão dos algoritmos de classificação está ligada à qualidade e sofisticação do modelo de dados. De acordo com os resultados, o atributo mais valioso para a previsão de desistência dos alunos consiste no número de aplicações de exames para os cursos de matemática e de programação (Vlatko Nikolovski, 2014).

O modelo de técnicas de avaliação apresentado pelo autor (Vlatko Nikolovski, 2014) aponta para duas grandes melhorias que podem ser notadas. A primeira observação, consta que o agrupamento de cursos em subconjuntos baseados no campo de estudo traz uma grande melhoria no processo de *data mining* e, a segunda e última observação, aponta para a qualidade e o tamanho do conjunto de dados dos alunos. Além disso, os resultados seriam mais eficazes se o conjunto de dados analisado fosse maior e mais organizado.

3.3 PEAR

Este caso de estudo (Rocha, 2006) detalha as especificações da ferramenta PEAR, *Post-Processing Environment for Association Rules*, ferramenta que tem como finalidade implementar uma nova metodologia de pós-processamento de regras de associação, utilizando um conjunto de operadores, permitindo também visualização das regras, dando ao utilizador uma forma de navegar pelo conjunto de regras.

A ferramenta PEAR contém os seguintes objetivos:

- Reduzir o tempo de aprendizagem do utilizador, utilizando uma interface simples;
- Funcionamento da ferramenta em múltiplas plataformas;
- Ler ficheiros de dados que contém regras de associação;
- Utilização de operadores para navegar no espaço de regras de associação.

O caso de estudo tem como finalidade melhorar a ferramenta PEAR, apresentando uma nova ferramenta, com uma nova interface, mais evoluída, rápida e capaz de incorporar facilmente novas formas de navegação, o PEAR-R.

A grande diferença é a utilização do R na ferramenta PEAR, que visa refletir sobre as suas principais potencialidades e limitações, melhorando-a. O PEAR-R opta por utilizar as funções de geração de gráficos do R.

Após terem finalizado a ferramenta PEAR-R, esta foi avaliada por um pequeno grupo de utilizadores com conhecimentos na área de informática e com conhecimentos na extração de dados, no qual concluíram que, de uma forma geral, a opinião dos inquiridos foi favorável quanto à utilização do PEAR-R ao invés do PEAR.

4 Extração de conhecimento de dados

Neste capítulo é apresentado o estudo que foi feito ao longo do projeto sobre *data mining* e técnicas de *data mining*, as regras de associação, os algoritmos mais relevantes que extraem regras de associação e a metodologia de pós-processamento de regras de associação.

4.1 Data Mining

Data mining é o processo analítico para explorar grandes quantidade de dados, com o objetivo de descobrir padrões e estabelecer relações para resolver problemas através da análise de dados (Kamber, 2006). Os padrões podem ser extraídos através de algoritmos de *data mining* que geram regras de associação. As regras de associação permitem caracterizar o quanto a presença de um conjunto de atributos implica a presença de um outro conjunto de atributos.

A análise de dados educacionais é um conjunto de métodos computacionais, psicológicos e de pesquisa, que permite analisar os dados obtidos durante todo o processo de aprendizagem do aluno, desde os serviços oferecidos pela instituição até à aprendizagem.

A utilização de técnicas de *data mining é* um passo particular no processo de descoberta de conhecimento em bases de dados (KDD), envolvendo a aplicação de algoritmos específicos para extrair padrões (modelos) de dados (Kamber, 2006). As etapas adicionais, como a preparação de dados, seleção de dados, transformação de dados e interpretação adequada dos resultados da exploração de dados, garante que o conhecimento adquirido é derivado dos dados analisados. *Data mining* oferece recursos para processar dados de diferentes tipos (qualitativamente e quantitativamente) e com origens de diversas fontes.

O objetivo de um estudo de *data mining* é criar um modelo descritivo ou um modelo preditivo. Um modelo descritivo apresenta as principais características do conjunto de dados. A tarefa de modelagem descritiva característica é o agrupamento. O propósito de um modelo preditivo é permitir que o *data miner*, ferramenta que aplica os algoritmos aos dados, preveja um valor desconhecido (muitas vezes futuro) de uma variável específica (Rustemi & Halili, 2016).

A exploração de dados já foi aplicada em diferentes áreas, incluindo vendas, bioinformática e luta contra o terrorismo (Maksood & Achuthan, 2006). Nos últimos anos tem havido um crescimento

no interesse do uso de estudos d*ata mining* para investigar questões científicas na área da educação, esta área é denominada como *educational data mining*.

A área de *data mining* na educação tem como objetivo prever o comportamento na aprendizagem dos alunos, criar e melhorar modelos de domínio, estudar os efeitos do apoio aos estudantes e aperfeiçoar o conhecimento científico sobre a aprendizagem dos estudantes (Romero, et al., 2007).

O processo de *data mining* aplicado na educação visa a criação de métodos de exploração de dados provenientes de dados educacionais e a utilização desses métodos para melhorar o comportamento dos alunos na sua aprendizagem e os métodos pelos quais eles aprendem (Romero, et al., 2007).

4.1.1 Técnicas de *Data Mining*

Existem várias abordagens de *data mining* para desenvolver padrões dos quais, algoritmos de regressão, regras de associação e algoritmos de agrupamento. Estes são usados para extrair conhecimento dos dados.

De acordo com Yongjian Fu (Fu, s.d.) existem vários métodos de exploração de dados presentes nas seguintes categorias:

- **Classificação**

  Tem como objetivo desenvolver um modelo que pode inferir uma varável dos dados de uma combinação de outros aspetos de dados. Requer *tags* para as variáveis de saída de um conjunto de dados limitado, onde um rótulo representa algumas informações confiáveis sobre a varável de saída. Em alguns casos é preciso considerar o grau para o qual o rótulo é confiável ou não. Este método pode ser usado para estudar as características de um modelo importante para a previsão, dando informações na construção subjacente e, também, pode ser usado para prever o valor de saída onde não é desejável obter uma *tag* para essa construção.

- **Agrupamento**

  No agrupamento o objetivo é encontrar pontos nos dados que se agrupam naturalmente, dividindo o conjunto de dados em pequenos subconjuntos, chamados *clusters.* É particularmente útil nos casos em que as categorias mais comuns dentro do conjunto de dados não são previamente conhecidas. Os *clusters* podem ser criados em diferentes tamanhos, por exemplo, as escolas podem ser agrupadas num conjunto e os alunos poderiam ser agrupados

num outro conjunto, para investigar semelhanças e diferenças entre estudantes, ou então, as ações dos alunos podem ser agrupadas para investigar padrões de comportamento.

- **Relação entre os dados**

Tem como objetivo descobrir relações entre variáveis, num conjunto de dados com um grande número de variáveis, tentando assim encontrar as variáveis que contêm uma forte conexão associada a uma varável de interesse particular ou descobrir qual a relação entre as varáveis mais fortes. Existem quatro tipos de relacionamento, dos quais:

i. **Regras de Associação -** contém o objetivo de encontrar regras "if-then" de forma a encontrar um conjunto de valores e atribuir a esses valores uma variável específica. Por exemplo, se o aluno é frustrado então o estudante tem um objetivo mais forte de aprender do que o objetivo de desempenho (ver secção 4.2).

ii. **Correlação -** contém o objetivo de encontrar correlações lineares, de carácter positivo ou negativo, entre as variáveis.

iii. **Padrões sequenciais –** o objetivo é encontrar associações temporais entre eventos.

iv. **Dados causais –** o objetivo é descobrir se um evento foi a causa de outro evento, utilizando a informação do motivo que desencadeou o evento.

Por norma, os dados são divididos em dois conjuntos, conjunto de treino e conjunto de testes. O treino é usado para treinar os modelos e o conjunto de testes são usados para verificar a precisão do modelo resultante (Ozer, 2008).

Nos últimos anos, os métodos de dados educacionais permitiram expansão na sofisticação dos modelos, em particular, esses métodos fizeram inferir o ensino superior sobre o comportamento dos estudantes. Estes modelos enriqueceram a capacidade de prever o conhecimento dos estudantes e desempenho futuro, incorporando modelos de previsão (Jacob, et al., 2015).

4.2 Regras de associação

Esta técnica de *data mining* permite descobrir a presença de conjuntos de elementos nos registos analisados (Kumbhare & Chobe, 2014). Baseado num exemplo: “60% dos clientes que compram o livro X e Y também compram o livro Z”. O objetivo de extrair regras de associação é formar regras a partir de conjuntos de elementos que aparecem juntos nas mesmas transações.

Uma regra de associação é composta por dois conjuntos de *items*: um antecedente ou lado esquerdo (LHS) e um consequente ou lado direito (RHS) e são representadas da seguinte forma de Antecedente -> Consequente.

De seguida são apresentadas medidas de avaliação de interesse das regras de associação, começando pelos conceitos mais importantes, suporte e confiança (Kamber, 2006).

4.2.1 Medidas de Avaliação Objetivas

As medidas de avaliação objetivas, dependem exclusivamente da estrutura das regras extraídas e dos dados utilizados no processo de extração de regras. Uma forma de descobrir regras interessantes através do recurso a medidas objetivas é definir, por exemplo, quando essa regra é interessante ou não, com recurso ao suporte e à confiança. De salientar que estas medidas, embora sejam úteis, não conseguem capturar toda a complexidade do processo de descoberta de padrões, uma vez que apenas analisam a força estatística das regras.

Ao longo do tempo foram criadas outras medidas estatísticas que podem ser utilizadas para avaliar as regras de associação extraídas. A correlação, por exemplo, analisa a força do relacionamento do conjunto enquanto que, por sua vez, o teste do qui-quadrado é uma medida que testa a independência e a correlação entre os *items* da regra de associação (Lenca, et al., s.d.).

4.2.1.1 Suporte e Confiança

O suporte determina a frequência com que um conjunto de elementos ocorre em todas as transações. Assim, para uma determinada regra de associação {X} -> {Y}, o suporte da regra mede o número total de registos de elementos que contém os conjuntos de itens X e Y.

$$Suporte\ (X \rightarrow Y) = \frac{Frequência\ de\ X\ e\ Y}{Total\ de\ Transações} \quad (1)$$

O numerador diz respeito ao número de transações em que X e Y ocorrem simultaneamente e o denominador refere-se ao número total de transações do conjunto de dados analisado.

A confiança mede a força da regra, ou seja, a percentagem de transações em que o antecedente contém o consequente.

$$Confiança\ (X \rightarrow Y) = \frac{Suporte\ (X \cup Y)}{Suporte\ (X)} \quad (2)$$

O numerador refere-se ao número de transações em que X e Y ocorrem simultaneamente. O denominador refere-se à quantidade de transações em que o item X ocorre. Em termos gerais a confiança mede a probabilidade condicional de ocorrer Y dado que ocorreu X.

De seguida é apresentado um exemplo de um conjunto muito pequeno de transações e denominado o seu suporte e confiança (Kamber, 2006).

Transações. ID	Produtos comprados
T1	leite, amendoins, fraldas
T2	café, amendoins, cerveja
T3	leite, café, amendoins, cerveja
T4	café, cerveja

Tabela 1 - Conjunto de Transações

Do conjunto anterior de transações podemos apresentar a seguinte regra:

R1 - café, amendoins -> cerveja

Suporte = 50% e Confiança = 100%

No exemplo dado, apesar do valor da confiança ser igual a 100%, não é suficiente para determinar se a regra anterior é válida ou não, uma vez que o suporte não é suficiente alto.

4.2.1.2 Interesse das regras

Como o processo de *data mining*, geralmente, é efetuado a grandes quantidades de dados, isto pode indicar que a quantidade de regras de associação também seja vasta, o que pode induzir o utilizador a uma falsa conclusão, se analisar somente o suporte e a confiança. Para não ocorrer esta falsa conclusão, é preciso determinar o grau de interesse das regras obtidas (Geng & Hamilton, 2006).

O *lift* ou coeficiente de interesse é uma medida para avaliar dependências entre o antecedente e o consequente da regra. A sua variação é entre 0 e infinito e consiste em: quanto maior o valor do *lift*, mais interessante é a regra, pois maior é a dependência entre os itens que a constituem.

O *lift* de uma regra é dado pela seguinte fórmula:

$$Lift\ (X \rightarrow Y) = \frac{confiança(X \rightarrow Y)}{suporte(Y)} = \frac{suporte(X \cup Y)}{suporte(X) * suporte(Y)} \quad (3)$$

Se $Lift\ (X \rightarrow Y) = 1$, então X e Y são independentes.

Se $Lift\ (X \rightarrow Y) > 1$, então X e Y são positivamente dependentes.

Se $Lift\ (X \rightarrow Y) < 1$, então X e Y são negativamente dependentes.

Assim, se *lift* assumir o valor 1, existe independência entre X e Y. Valores para *lift* que se afastam de 1 indicam que a evidência de X fornece informação sobre Y e as regras começam a ter uma maior relevância de informação acerca da relação entre elas.

4.2.2 Medidas de Avaliação Subjetivas

Enquanto as medidas de avaliação objetivas dependem apenas dos dados e medem estatisticamente a forma das regras, as medidas de avaliação subjetivas dependem do conhecimento e interesse dos utilizadores (Sultan, 2004).

Ahmed Sultan conclui três critérios subjetivos, sendo os dois primeiros os mais importantes, utilidade, previsibilidade e novidade.

- **Previsibilidade e Utilidade**

Uma regra é inesperada se não for conhecida previamente pelo analista ou se for oposta ao conhecimento que o mesmo contém, surpreendo assim o analista. Uma regra é considerada útil se trouxer um conhecimento que é utilizado na decisão.

Baseado nestes dois critérios subjetivos temos os seguintes cenários possíveis:

i. Regras que são úteis e inesperadas;
ii. Regras que são inesperadas e não úteis;
iii. Regras que são esperadas e úteis.

A previsibilidade e utilidade tem de ser endereçadas individualmente e representadas separadamente na avaliação subjetiva. Se uma regra é inesperada e não tem utilidade, então não interessa tanto como uma regra inesperada e útil. No entanto ambas as regras deverão de ser apresentadas, mas com diferentes graus de interesse.

A utilidade de uma regra pode ser obtida por diferentes algoritmos de *data mining*, isto é, um conjunto de regras extraídas geradas por um algoritmo particular pode ser mais útil do que um conjunto de regras extraídas por um diferente algoritmo. Ahmed (Sultan, 2004) conclui que a utilidade é uma medida de avaliação que tem de ser avaliada independente da previsibilidade.

- **Novidade**

A novidade, no processo de pré-processamento, pode ser usada como um filtro para selecionar e concentrar um conjunto de elementos que deverão de ter mais em atenção.

No processo de pós-processamento, este método de avaliação consegue analisar a descoberta de conhecimento objetivo e subjetivo de forma limitar o número de regras a extrair que são mais fáceis de analisar pelo analista.

Apesar de Sultan acreditar que este método de avaliação subjetiva não contém nenhuma prova concreta que traz valor de interesse no processo de *data mining* (Sultan, 2004), existe um artigo de autores como Sugato Basu e Raymond Mooney que utiliza de forma eficiente este método de avaliação para definir uma estimativa de distância entre pares de palavras (Basu, et al., 2001).

4.3 Algoritmos de extração de regras de associação

Nesta secção são apresentados alguns algoritmos para extração de regras de associação. Em particular, serão apresentadas algumas noções sobre os seguintes algoritmos: Apriori, AIS, SetM, FP-Growth, AprioriTID e Apriori Hybrid. No final da secção é apresentada uma tabela de comparação de todos os algoritmos analisados.

4.3.1 AIS

O algoritmo AIS foi o primeiro algoritmo proposto por Agrawal, Imielinski e Swami (Agrawal, et al., 1993) para extrair regras de associação. Este algoritmo foca-se em melhorar a qualidade das bases de dados com a funcionalidade das decisões de suporte. Neste algoritmo apenas é gerado um elemento consequente de regras de associação, o que significa que o consequente das regras só contém um elemento, ou seja, regras X∩Y->Z podem ser geradas, ao contrário das regras X->Y∩Z.

No algoritmo AIS, os *items* candidatos são gerados e contabilizados *on-the-fly*, ou seja, conforme a base de dados é explorada. Por cada transação, é determinado qual dos *itemsets* do passo anterior estão presentes na corrente transação e os novos *itemsets* candidatos são gerados pela extensão destes *itemsets* por outros elementos presentes na transação (Figura 2).

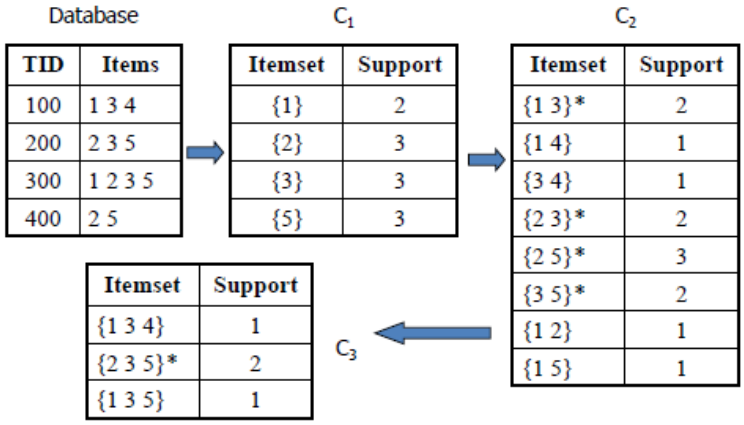

Figura 2 - Aplicação do algoritmo AIS (Sayad, 2008)

Este algoritmo origina a criação de *itemsets* desnecessários, que no final do processo são considerados desnecessários, por terem um comprimento reduzido, o que faz com que ocupe muito espaço e desperdice o esforço efetuado (Agrawal, et al., 1993).

4.3.2 SetM

No algoritmo SetM (Kumbhare & Chobe, 2014), os *itemsets* candidatos são gerados *on-the-fly*, à medida que a base de dados é explorada, no entanto, só são contabilizados no final do processo. Os novos *itemsets* candidatos são gerado da mesma forma que o algoritmo AIS, mas o identificador da transação (TID) gerada é guardado com o *itemset* candidato numa estrutura sequencial. No final do processo, o valor do suporte é determinado com base na ordenação e agregação da estrutura sequencial, previamente criada (Kumbhare & Chobe, 2014).

Contém a desvantagem de que para cada *itemset* candidato existem múltiplas entradas com um valor de suporte diferente e, pelas suas semelhanças com o algoritmo AIS, também apresenta a mesma desvantagem do que este, enunciada acima.

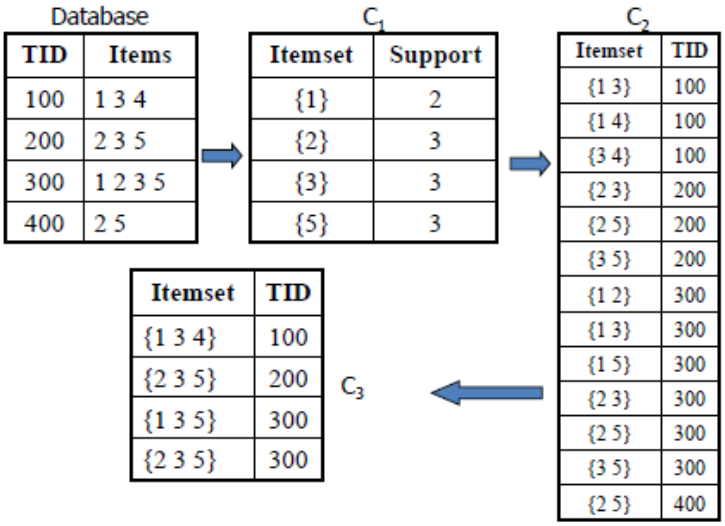

Figura 3 - Aplicação algoritmo SetM (Sayad, 2008)

4.3.3 Apriori

O algoritmo Apriori (Kumbhare & Chobe, 2014) é o algoritmo mais usado para extração de regras de associação e difere dos algoritmos AIS e SetM na maneira pela qual cada *itemset* candidato é gerado e contado.

Neste algoritmo os *itemsets* candidatos a serem contabilizados numa dada iteração são gerados a partir dos *itemsets* frequentes gerados na iteração anterior, sem considerar as transações na base de dados analisada. Este algoritmo beneficia do facto de que para qualquer subconjunto de um *itemset* frequente é também ele frequente. Com base neste conceito, os *itemsets* candidatos que contêm K-*itemsets*, podem ser gerados combinando (K-1) *itemsets*, seguido da remoção de todos os subconjuntos que não são frequentes. Este procedimento faz com que o número resultante de *itemsets* candidatos seja muito menor que os outros algoritmos previamente apresentados.

A Figura 4, apresenta o algoritmo Apriori, onde é possível visualizar que o algoritmo faz K iterações na base de dados, onde K é o tamanho do maior *itemset* frequente. Esta é a maior desvantagem deste algoritmo, especialmente se K for substancialmente grande. Apesar disso, este algoritmo inspirou a comunidade a procurar um melhor algoritmo de forma a ultrapassar este problema que originou várias otimizações, como o caso do AprioriTD e AprioriHybrid (Kumbhare & Chobe, 2014).

Algorithm 1 Apriori algorithm

```
1: begin
2:     L_1 ← Frequent1 − itemset
3:     k ← 2
4:     while L_{k−1} ≠ φ do
5:         Temp ← candidateItemSet(L_{k−1})
6:         C_k ← frequencyOfItemSet(Temp)
7:         L_k ← compareItemSetWithMinimumSupport(C_k, minsup)
8:         k ← k + 1
9:     end while
10:    return L
11: end
```

Figura 4 - Algoritmo Apriori (Kim, 2014)

Na figura seguinte é possível visualizar um exemplo do algoritmo Apriori (Figura 5).

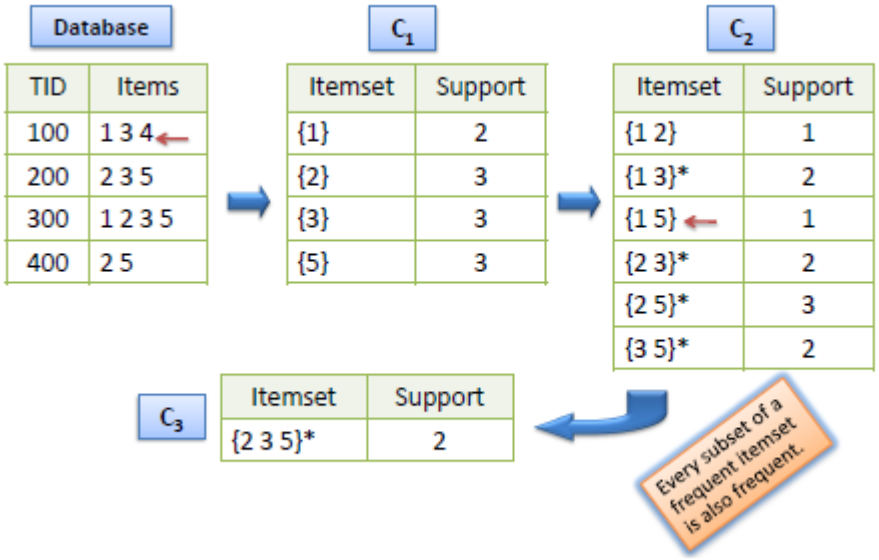

Figura 5 - Aplicação algoritmo Apriori (Sayad, 2008)

4.3.4 Apriori TID

A principal diferença do algoritmo AprioriTID (Kumbhare & Chobe, 2014) relativamente ao Apriori é que este algoritmo não utiliza a base de dados para determinar o valor do suporte nos *itemsets* candidatos após a primeira iteração. Para determinar esse valor, estabelece uma codificação baseada nos *itemsets* candidatos do passo anterior. Esta codificação é armazenada no conjunto Ck

em que cada membro está na forma de <Tid,{Xk}>, onde Xk é um potencial K-*itemset* na transação com identificador TID.

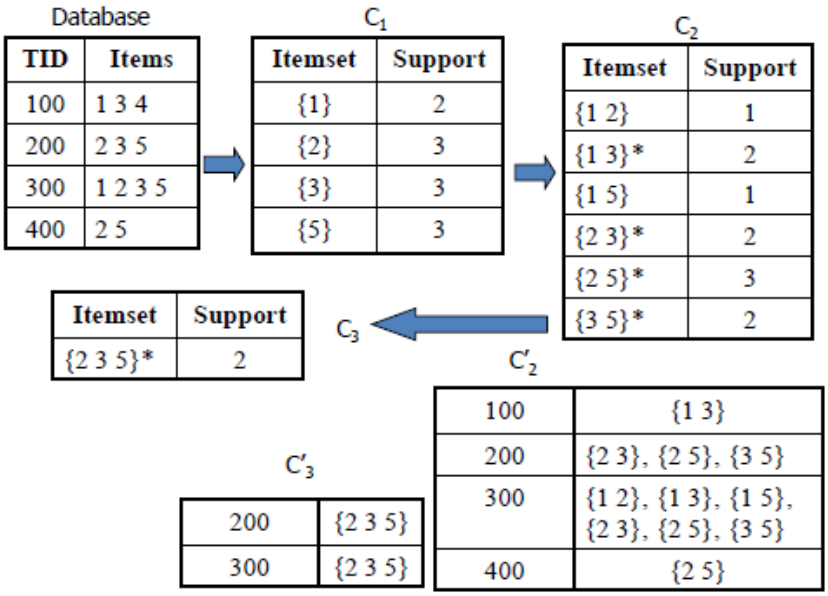

Figura 6 - Aplicação algoritmo Apriori TID (Sayad, 2008)

Comparando o algoritmo Apriori e o AprioriTID, verifica-se que este último tem um desempenho superior quando o conjunto de candidatos não excede a memória da base de dados, perdendo assim a sua vantagem pelo Apriori se o conjunto de candidatos exceder a memória da base de dados.

4.3.5 Apriori Hybrid

Como o algoritmo Apriori contém uma melhor performance nas primeiras iterações e o algoritmo AprioriTID contém uma melhor performance nas iterações seguintes, um novo algoritmo foi desenhado, Apriori Hybrid (Kumbhare & Chobe, 2014), que conjuga os dois algoritmos descritos acima, utilizando as características de ambos. Nas primeiras iterações usa o algoritmo Apriori e, posteriormente, nas iterações seguintes usa o algoritmo AprioriTID (Kumbhare & Chobe, 2014).

Contudo o momento de transação do algoritmo Apriori para o algoritmo AprioriTID não é linear.

4.3.6 FP-Growth

O algoritmo FP-Growth (Verhein, 2008) utiliza a estratégia de dividir para conquistar. Primeiro, requer a compressão de dados numa FP-*tree*, que armazena a informação dos *itemsets* frequentes. De seguida, cada *itemset* frequentes possui, a partir de uma FP-*tree*, uma estrutura especial comprimida de todos os possíveis *itemsets* que podem ser formados com o *itemset* em questão. Desta forma, sucessivas divisões do conjunto de dados, são realizadas separadamente para descobrir os *itemsets* frequentes.

Conclui-se que este algoritmo contém duas etapas importantes, a construção da FP-*tree* e a mineração dos *itemsets* frequentes sobre a FP-*tree* construída.

Uma FP-*tree* é uma estrutura compacta que guarda informação sobre os *itemsets* frequentes, presentes nos dados analisados e é construída da seguinte forma:

- Na primeira passagem, depois de se encontrar os *itemsets* frequentes, é preciso encontrar o suporte para cada *itemset* e ordenar os *itemsets* frequentes de forma descendente baseado no suporte de cada *itemset*;
- Na segunda passagem, é feita a leitura de cada transação e esta é mapeada por um caminho na FP-*tree*. Como diferentes transações podem conter muitos *itemsets* em comum, os caminhos podem se sobrepor. Quanto maior for a existência de caminhos sobrepostos, maior é a compressão alcançada na FP-*tree* construída.

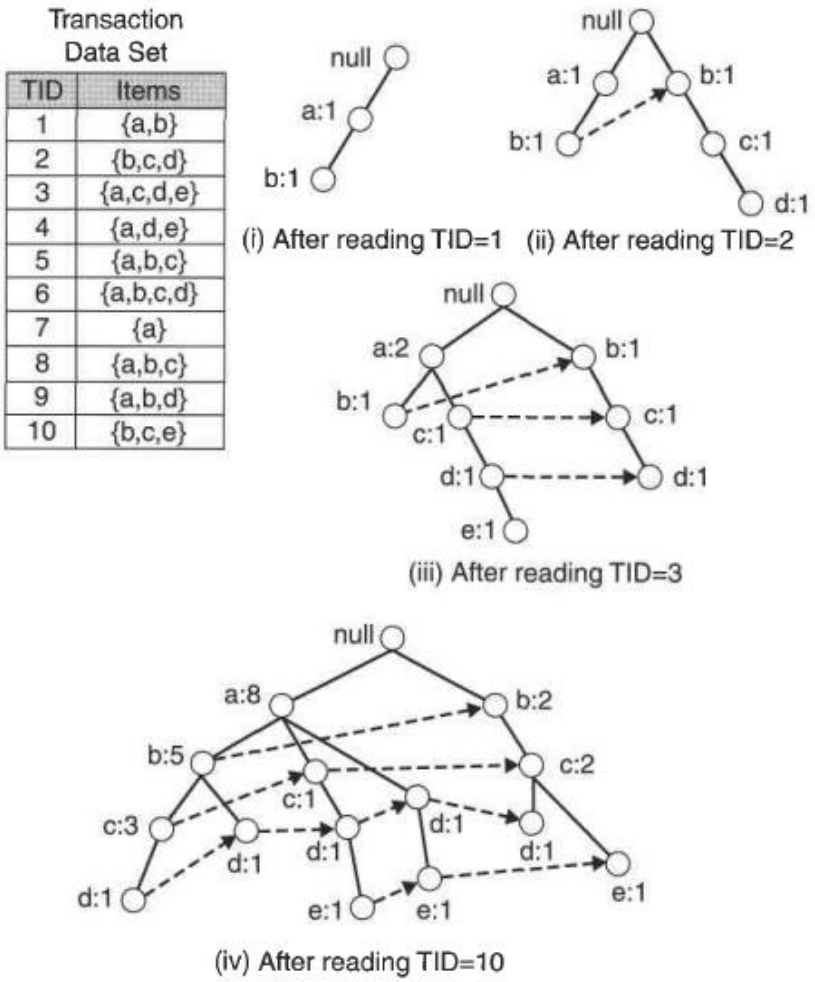

Figura 7 - Exemplo de uma FP-t*ree* (Verhein, 2008)

Depois da construção de uma FP-*tree*, é preciso extrair os *itemsets* frequentes da FP-*tree* (Kumbhare & Chobe, 2014). Este processo é executado através do algoritmo FP-Growth.

O algoritmo FP-Growth explora a FP-*tree* de baixo para cima. Visto que cada transação é mapeada por um caminho da FP-*tree*, é possível explorar os *itemsets* frequentes que terminam com um item particular, analisando somente os caminhos que contenham um *item* em particular.

Baseado no exemplo dado na Figura 7, o algoritmo FP-Growth iria extrair primeiro os elementos terminados em E, depois por D, C, B, terminando em A. O passo inicial passaria por descobrir todos os caminhos que incluem o elemento E, a estes primeiros caminhos são denominados *prefix path*. O valor de suporte final é calculado adicionando os valores de suporte associados com o elemento em particular. Suponhamos que o valor de suporte do elemento E é 6 e o valor do mínimo de suporte é 3, então o elemento E é frequente. Como este elemento é frequente, o algoritmo

resolverá os sub-problemas de encontrar *itemsets* frequentes que acabam em AE, BE, CE e DE. Para resolver os sub-problemas o algoritmo cria várias FP-*tree* condicionais necessárias para encontrar os *itemsets* que acabem com as combinações anteriores, antes de passar para os elementos seguintes.

No artigo de Kumbhare (Kumbhare & Chobe, 2014), o autor conclui que o algoritmo FP-Growth contém uma maior vantagem relativamente aos outros algoritmos analisados, como se pode ver na tabela seguinte.

Características	AIS	SetM	Apriori	AprioriTID	Apriori Hybrid	FP-Growth
Suporte de dados	Pequeno	Pequeno	Limitado	Grande	Muito Grande	Muito grande
Velocidade inicial	Lento	Lento	Rápido	Lento	Rápido	Rápido
Velocidade final	Lento	Lento	Lento	Rápido	Rápido	Rápido
Precisão	Muito Impreciso	Impreciso	Impreciso	Mais preciso que o Apriori	Mais preciso que AprioriTID	Mais preciso

Tabela 2 - Comparação entre algoritmos (Kumbhare & Chobe, 2014)

Contudo para este trabalho foi usado o algoritmo Apriori, pois a própria ferramenta RStudio contém uma biblioteca chamada 'arules' que utiliza o algoritmo Apriori.

4.4 Pós-Processamento de regras de associação

O pós-processamento de regras de associação consiste na simplificação, avaliação e visualização das regras extraídas. O pós-processamento pode ser realizado através de ferramentas desenvolvidas para tal ou pela análise de especialistas, de modo a permitir selecionar (de um grande conjunto de regras) as regras mais importantes. Uma ferramenta de pós-processamento de regras de associação deve apresentar as regras extraídas, os seus valores de suporte, confiança e *lift*, permitindo também a navegação no conjunto das regras (Domingues & Rezende, 2005).

Por norma, em grandes quantidades de dados, os algoritmos de regras de associação geram um elevado número de regras de associação, dos quais muitas delas podem não ter significado ou interesse para o objetivo do caso de estudo. Por este motivo, é importante fazer uma análise cuidada das regras de associação extraídas e apresentar um número bastante reduzido de regras e com relevância para o utilizador ou para o caso de estudo em questão.

No artigo de Marcos Domingues (Domingues & Rezende, 2005) os autores definem as seguintes etapas do processo de pós-processamento das regras de associação:

1. **Filtragem de Conhecimento**

É realizado por árvores de decisão ou truncagem de regras. Os algoritmos produzem árvores de decisão com muitos caminhos (folhas) ou criam muitas regras de decisão muito especificas, cobrindo poucos exemplos. Para contornar este problema pode-se restringir os atributos ou ordenar as regras por meio de métricas (medidas de avaliação), já descritas na secção 4.2.1.

2. **Interpretação e Explanação**

É aplicado quando o conhecimento obtido é utilizado por um utilizador ou numa ferramenta de pós-processamento. O conhecimento adquirido pode ser documentado, visualizado e modificado de modo a ser mais compreensível para o utilizador. Também é possível verificar se o conhecimento adquirido está de acordo com o conhecimento que o utilizador contém previamente.

3. **Avaliação**

Para esta etapa existem alguns critérios específicos, tais como: compreensibilidade, complexidade computacional e grau de interesse.

4. **Integração do Conhecimento**

Os sistemas tradicionais de suporte à decisão são dependentes de uma única técnica, estratégia e modelo. Os novos e mais sofisticados sistemas de suporte à decisão permitem combinar ou refinar os resultados obtidos de vários modelos, obtendo uma maior precisão e um conhecimento mais exato.

5 Tecnologias utilizadas

Neste capítulo são apresentadas as tecnologias utilizadas para o desenvolvimento do projeto, onde é contextualizado o porquê do seu uso.

Algumas das tecnologias mencionadas serão alvo de um detalhe mais pormenorizado, permitindo ao leitor conhecer o mínimo necessário para compreender as descrições técnicas ao longo do relatório.

Em seguida é apresentada uma tabela síntese com as tecnologias utilizadas:

Área	Tecnologias
Desenvolvimento	Shiny, R, Markdown,
Ferramentas	RStudio, Microsoft IIS, Shiny Server

Tabela 3 - Tecnologias utilizadas

5.1 R

O R é uma linguagem de programação usada para análise estatística e algoritmos de *machine learning* similar à linguagem e à interface do S (Hornik, 2017).

Tal como S, o R permite que os utilizadores adicionem funcionalidades à ferramenta para os outros utilizadores, definindo funções/bibliotecas. O R é facilmente personalizado com a implementação de bibliotecas, criadas pela comunidade desta ferramenta.

O R é um ambiente para realizar operações estatísticas e gerar relatórios de análise de dados em formatos gráficos ou de texto. Os comandos R inseridos na consola são avaliados e executados. Uma das limitações do R é o facto de não conseguir manipular determinados caracteres de formatação automática, tais como '-'.

O R contém as seguintes características (Hornik, 2017):

- Manipulação de dados;
- Contém uma linguagem interpretada, ou seja, acessível através da linha de comandos;
- Possui facilidade de armazenamento e manuseio de dados;

- Técnicas gráficas para análise de dados;
- Suporta um conjunto de operadores para executar operações em matrizes.

A componente mais interessante do R, é a quantidade de gráficos que este possibilita gerar, tal como podemos visualizar alguns destes gráficos na Figura 8.

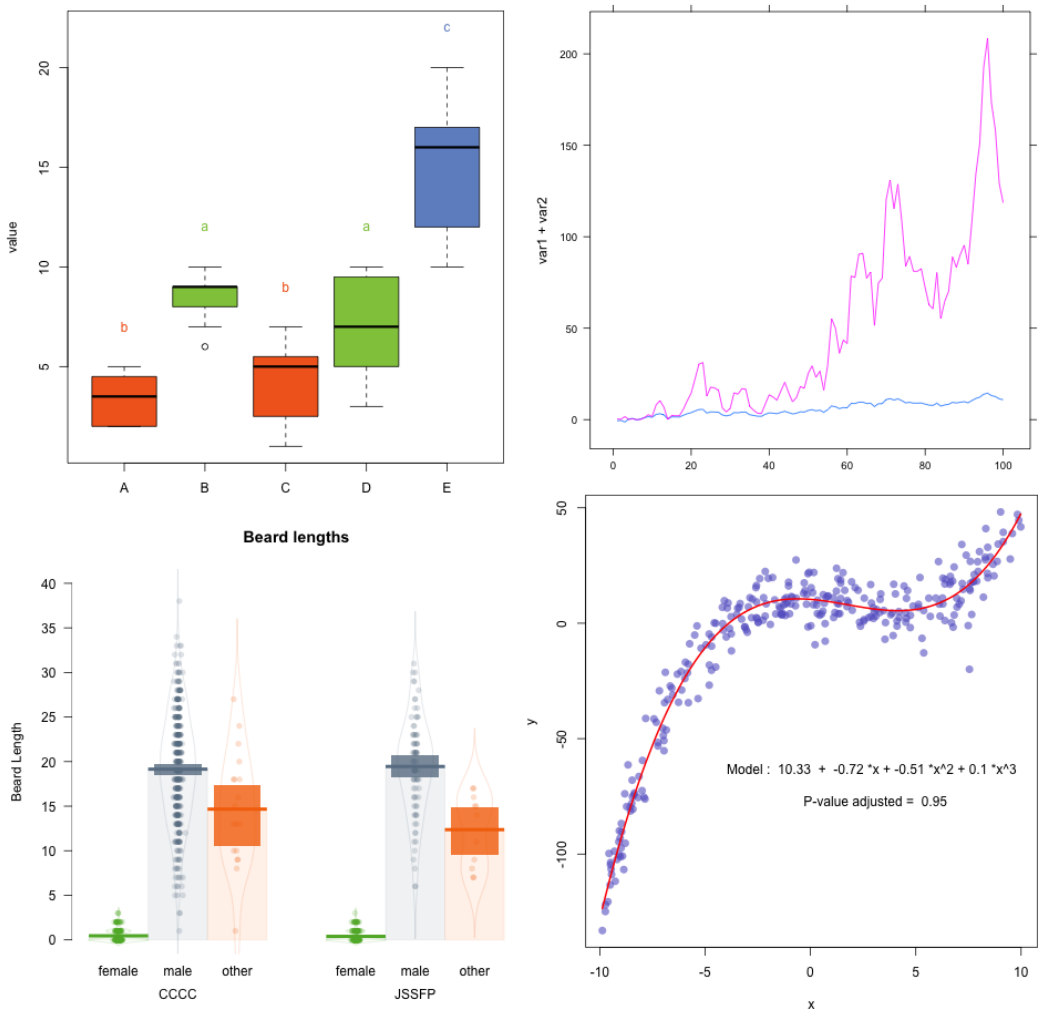

Figura 8 - Exemplos de gráficos do R (Maindonald, 2008)

5.2 Shiny

Shiny (Rstudio, Inc, 2017) é uma biblioteca que pode ser adicionada ao RStudio que facilita a criação de aplicações *web* com o R. Uma aplicação Shiny contém duas secções: interface gráfica e o código, executado no servidor que serve de *host* para a aplicação. A interface gráfica contém o código *front-end*, tais como, botões, separadores, gráficos, entre outros. Já o código do servidor

contém o código que executa por *back-end* como a manipulação de dados, recuperação de dados, geração de gráficos, entre outros.

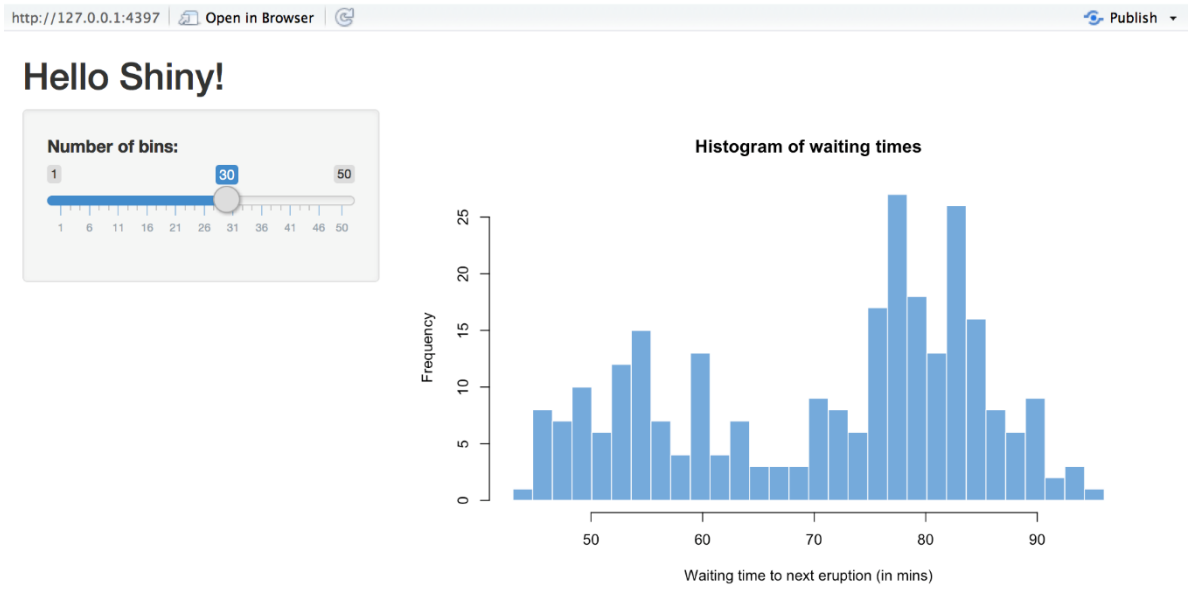

Figura 9 - Exemplo aplicação Shiny (Rstudio, Inc, 2017)

Shiny é uma biblioteca que não precisa de ferramentas adicionais, para além do RStudio, e providencia toda a análise avançada do R para utilizadores que desconhecem outra linguagem a não ser o R.

Todos os *scripts* em R podem ser utilizados no Shiny de forma a tornarem-se interativos. Cada aplicação Shiny pode ser alocada num servidor de forma a partilhar a aplicação com o mundo via *browser* ou localmente num computador pessoal, no qual só precisará do RStudio.

5.3 Markdown

A linguagem Markdown é uma linguagem simples de marcação criada por John Gruber, em 2004 (Gremberghe, 2016), usada para formatar elementos para documentos de texto, como *portable document format* (PDF).

Quando se escreve em Markdown, o texto é guardado num ficheiro simples de texto com a extensão .md ou .markdown que depois são formatados para outros tipos de ficheiros, através de uma aplicação capaz de converter e formatar a linguagem Markdown para texto.

No desenvolvimento da aplicação foi utilizado o R Markdown para fazer a conversão de um ficheiro Markdown para um documento HTML, capaz de converter as funções e os valores definidos na aplicação para valores visíveis no relatório.

O documento utilizado é um ficheiro R Markdown que contém dados provenientes da aplicação Shiny com extratos de código escritos em R.

Os documentos produzidos pela biblioteca R Markdown podem ser convertidos para outros tipos de ficheiro, como PDF ou Microsoft Word, sem qualquer desformatação do conteúdo deste.

5.4 RStudio

O RStudio, fundado por JJ Allare em 2010 (RStudio, Inc, 2015), tem como missão fornecer uma ferramenta *open source* para o ambiente de computação estatística R.

A ferramenta RStudio é escrit na linguagem de programação C++ e usa a ferramenta Qt (Qt Group, 1995), ambiente que permite projetar, desenvolver, implementar e manter o *software* multiplataforma de interface gráfica.

O RStudio (RStudio, Inc, 2011) está disponível em duas edições: o RStudio *Desktop*, no qual o programa é executado localmente como um aplicativo de desktop comum e o RStudio Server, que permite aceder ao RStudio utilizando um *browser* enquanto que o RStudio é executado num servidor remoto.

5.5 Microsoft IIS

O *Microsoft Internet Information Services* (IIS) (Wilson, s.d.) é um serviço *web* flexível e de uso geral da Microsoft, executado em sistemas operativos *Windows*, que permite hospedar aplicações para poderem ser interagidas por utilizadores através de um *browser* da *internet*.

O IIS funciona através de uma variedade de linguagens e protocolos padrão. A linguagem HTML é usada para criar elementos como texto, botões, posicionamentos de imagem, interações entre o utilizador e o servidor e hiperligações (Wilson, s.d.). O protocolo de transferência de hipertexto (HTTP, Hypertext Transfer Protocol) é o protocolo de comunicação básico usado para trocar informações entre servidores da *web* e os seus utilizadores (Fielding & Irvine, 1999). O protocolo

HTTPS - HTTP em SSL (Secure Sockets Layer) - usa o protocolo SSL para criptografar a comunicação e aumentar a segurança dos dados (Fielding & Irvine, 1999).

Um servidor *web* do IIS aceita solicitações de computadores clientes remotos e retorna a resposta apropriada. Essa funcionalidade básica permite que os servidores da w*eb* compartilhem e forneçam informações por meio de redes locais, como intranets corporativas e redes de longa distância, como a *Internet*. Um servidor w*eb* pode fornecer informações aos usuários em vários formulários, como páginas estáticas codificadas em HTML, através de trocas de arquivos como *downloads* e *uploads*, documentos de texto, arquivos de imagem e muito mais (Wilson, s.d.).

Os servidores IIS costumam ser usados como portais para aplicações sofisticadas, altamente interativas e baseados na *web*, que unem serviços *back-end* para criar sistemas de classe corporativa.

Com esta ferramenta é possível permitir a **FERA** num ambiente web onde poderá ser acedido por vários utilizadores, em simultâneo e através da *Internet.*

5.6 Shiny Server

Em alternativa ao Microsoft IIS existe o Shiny Server. O Shiny Server é uma componente do RStudio que permite implementar e executar as aplicações Shiny na máquina do utilizador ou a partir de um servidor.

Este programa cria um servidor *web* especificamente projetado para hospedar os aplicativos Shiny. Os aplicativos são hospedados num ambiente controlado, pois são hospedados numa máquina pessoal ou num servidor da organização. Com o Shiny Server é possível controlar o acesso dos utilizadores que entram na aplicação, contém também a habilidade de proteger as aplicações através do protocolo HTTPS e prioridade de suporte pelo RStudio (Rstudio, Inc, 2017).

Como não houve necessidade de implementar a aplicação desenvolvida num ambiente *web,* esta foi a alternativa implementada para processar a ferramenta.

6 Design da solução

Neste capítulo será apresentado cada um dos casos de uso. Para além de uma explicação, é apresentado os diagramas de sequência de cada caso de uso.

São também apresentados os requisitos funcionais e não funcionais através do modelo FURPS+ (Eeles, 2004), assim como a vista lógica e vista de implantação da **FERA**.

6.1 Requisitos Funcionais

Nesta secção são apresentados os requisitos funcionais da **FERA.** Para além de ilustrar o caso de uso a implementar é também apresentado uma descrição detalhada do mesmo e do seu diagrama de sequência.

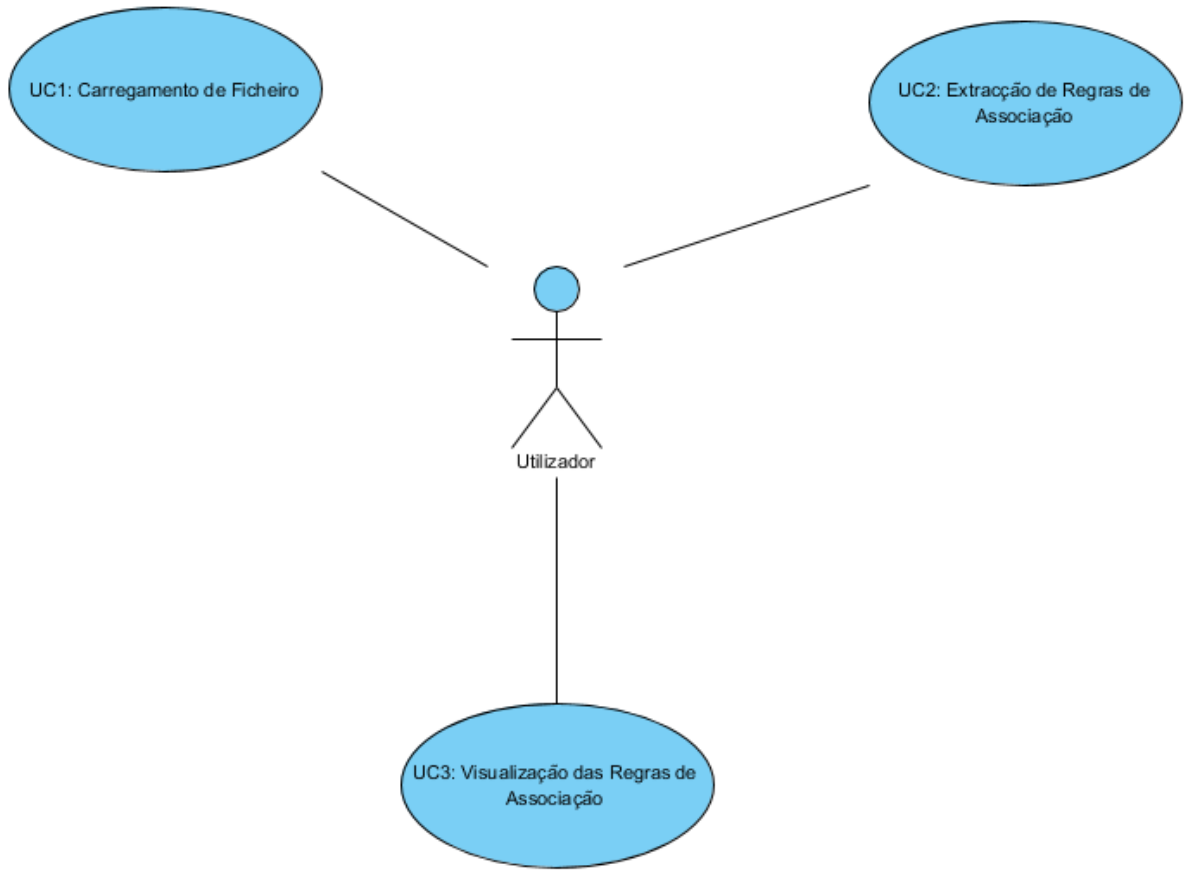

Figura 10 - Diagrama de Casos de Uso

Os casos de uso descrevem como é que os utilizadores interagem com a ferramenta desenvolvida. Durante as interações com a **FERA**, o utilizador envia pedidos para o sistema requerendo uma resposta do mesmo. Nas subsecções seguintes serão descritos detalhadamente cada um dos casos de uso.

6.1.1 UC1: Carregamento de Ficheiro

Neste caso de uso o utilizador pretende carregar um ficheiro na plataforma, que contém dados para gerar regras. O ficheiro tem de estar no formato XML ou CSV, de forma a ser aceite pela plataforma, caso contrário é retornado uma mensagem de erro para o utilizador.

Na figura seguinte (Figura 11) é ilustrado o diagrama de sequência do caso de uso em questão, seguido de uma descrição detalhada de cada passo.

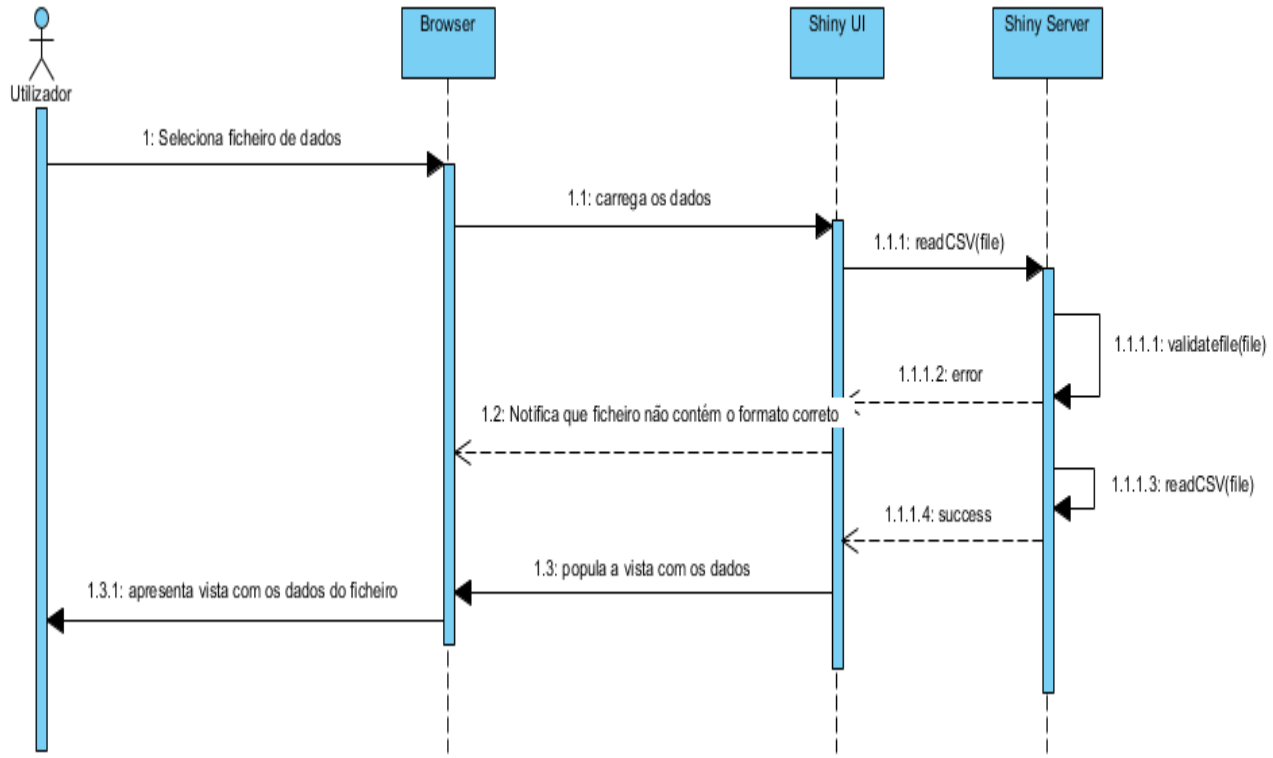

Figura 11 - Diagrama de sequência UC1

As interações entre os objetos intervenientes são as seguintes:

1. O utilizador seleciona o ficheiro de dados a carregar para a plataforma;
2. O pedido é registado e o ficheiro é carregado através das funções implementadas;
3. O R valida se o ficheiro é um ficheiro CSV ou um ficheiro XML;
4. Em caso de erro no passo anterior, é enviada uma mensagem para o utilizador a informar que o ficheiro não é permitido;
5. Em caso de sucesso do ponto 3 é executado o método readCSV, independentemente se for um ficheiro XML ou CSV;
6. Após leitura do ficheiro, os dados são carregados na plataforma onde o utilizador consegue visualizar os dados e verificar se o ficheiro carregado é o correto. Caso não seja o ficheiro correto, o utilizador contém a possibilidade de voltar a carregar um novo ficheiro, passando por todo o processo novamente.

6.1.2 UC2: Extração de Regras de Associação

No diagrama seguinte, da Figura 12, encontra-se representado o fluxo de ações do caso de "Extração de Regras de Associação" que visa gerar as regras para o utilizador de acordo com os parâmetros que este seleciona.

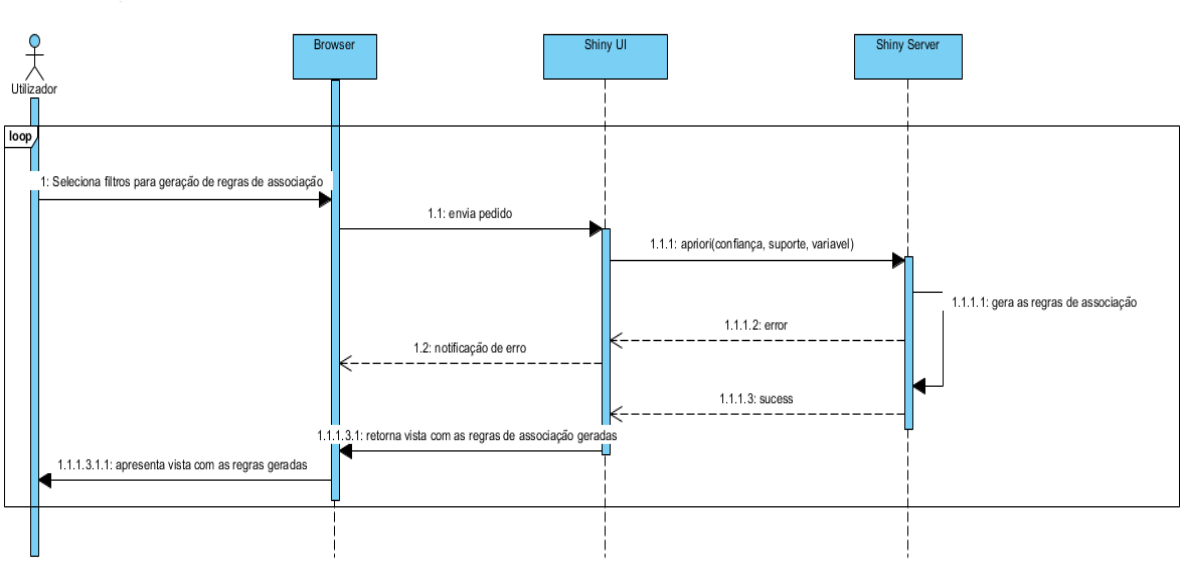

Figura 12 - Diagrama de sequência UC2

As interações entre os objetos intervenientes são as seguintes:

1. O utilizador seleciona os filtros para a geração das regras de associação. Os filtros a escolher são os seguintes:
    a. Confiança;
    b. Suporte;
    c. Variável.
2. O pedido é registado e enviado para o servidor;
3. O servidor tenta gerar as regras, baseada nos parâmetros escolhidos;
4. Em caso de erro, o utilizador é notificado através da janela de interface da ferramenta;
5. Em caso de sucesso, o utilizador visualiza as regras em forma de *array*;
6. Após geração das regras, o utilizador pode escolher novamente os parâmetros, voltando ao processo inicial do presente caso de uso.

6.1.3 UC3: Visualização das Regras de Associação

No diagrama seguinte encontra-se representado o fluxo de ações do caso de "Visualização das regras de associação" que tem como objetivo apresentar as regras geradas no caso de uso anterior e contém a possibilidade de fazer o *download* de um ficheiro HTML, das regras geradas e do gráfico gerado.

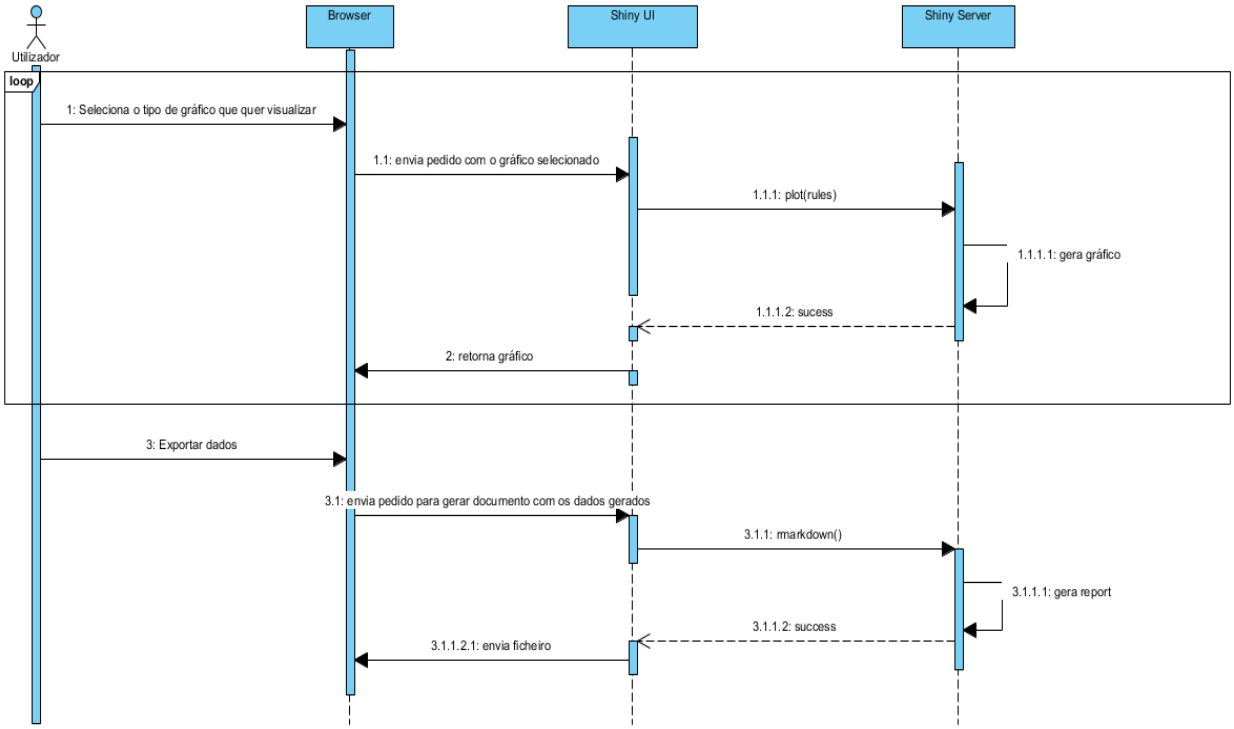

Figura 13 - Diagrama de sequência UC3

As interações entre os objetos intervenientes são as seguintes:

1. O utilizador seleciona o gráfico que deseja. As opções são as seguintes:
    a. Gráfico de dispersão;
    b. Sumário das regras.
2. O pedido é registado e enviado para o servidor;
3. O servidor cria o gráfico ou o sumário, através do método plot() e rendertable(), respetivamente;
4. Após ser efetuado o pedido do utilizador, é ilustrado ao utilizador pela interface gráfica da ferramenta;
5. O utilizador pode voltar a requisitar um outro pedido com outros parâmetros, reiniciando o processo no passo 2;
6. Após geração do gráfico/tabela, o utilizador pode exportar para um ficheiro de dados, selecionando o tipo de ficheiro que prefere;
7. O pedido é registado e enviado para o servidor;
8. Através do método rmarkdown() é gerado um ficheiro do tipo selecionado e descarregado na máquina do utilizador.

6.2 Requisitos não funcionais

Existem outros tipos de requisitos que necessitam de ser identificados, tais como usabilidade, suportabilidade, interface, restrições de *design* e/ou implementação. Para especificar essas componentes suplementares, utilizou-se o modelo FURPS+.

Requisitos Não Funcionais	Descrição
Funcionalidade	O tempo de resposta do sistema não deverá de ultrapassar 1 minuto
	O ficheiro de dados do utilizador deve ser intransmissível
Usabilidade	Interface interativa e acessível
	Notificações de aviso dirigidas ao utilizador devem ser informativas e construtivas
Desempenho	As ações e decisões de implementação devem ser pensadas para que o tempo de processamento das mesmas seja o mais curto possível
Suportabilidade	A plataforma deve ser funcional em *browsers* diferentes
Restrições de Implementação	A linguagem de programação utilizada é o R com o auxílio da framework RStudio e o *package* Shiny
Restrições de Interface	O sistema deve ser suportado pelos três principais *browsers* (Mozilla Firefox, Google Chrome e Microsoft Edge)
Restrições Físicas	Utilização de um servidor ou de uma máquina para implantação da solução

Tabela 4 - Requisitos Não Funcionais

6.3 Vista Lógica

A aplicação da ferramenta divide-se essencialmente em três camas:

- Camada de Persistência: responsável pela manutenção da informação tratada na camada de domínio;
- Camada de Domínio: contém a implementação das funcionalidades da plataforma;
- Camada de Apresentação: apresenta a informação gerada pela camada de domínio.

Na Figura 14 é ilustrada a vista lógica do sistema, apresentando uma perspetiva do mesmo.

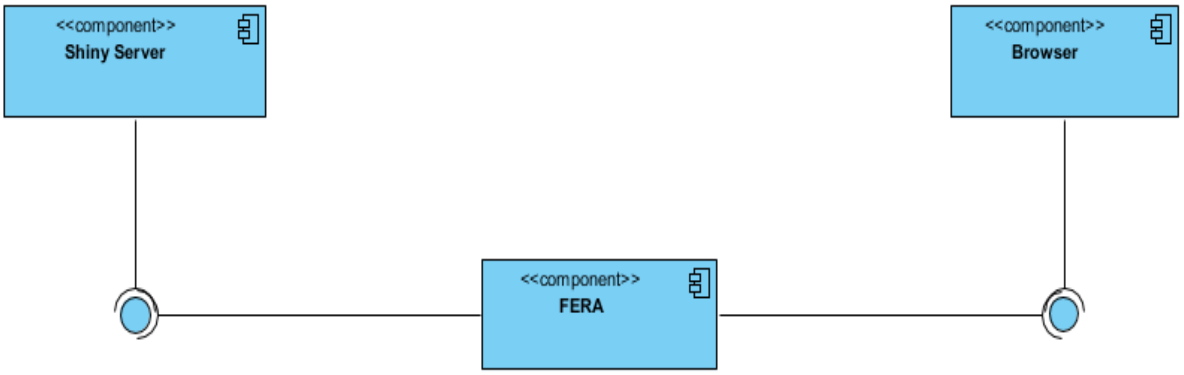

Figura 14 - Vista lógica

Neste diagrama encontram-se ilustrados os seguintes componentes, relevantes para a construção do sistema proposto:

- A componente **Shiny Server** representa a camada de persistência que a camada possui;
- a componente **FERA** representa a aplicação a ser desenvolvida;
- a componente **Browser** representa o cliente da aplicação desenvolvida.

6.4 Vista de Implantação

Na figura seguinte encontra-se ilustrada a vista de implantação, onde se encontra representado as diferentes máquinas do sistema.

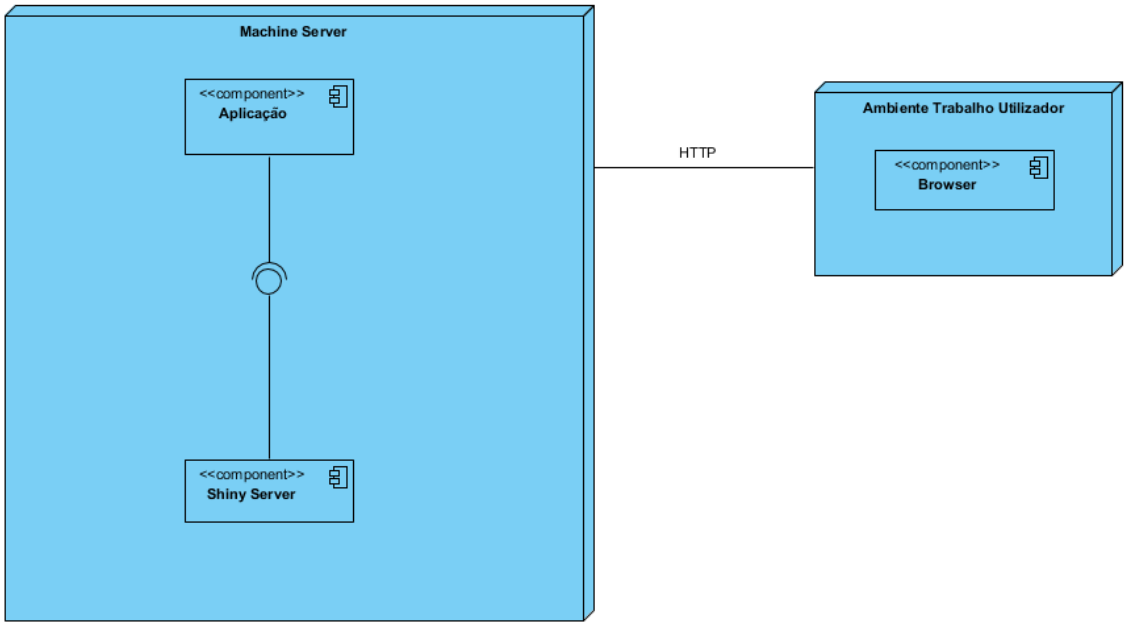

Figura 15 - Diagrama de Implementação

Como ilustrado na Figura 15, a arquitetura da aplicação baseia-se em duas máquinas, uma que contém a **Aplicação** e o **Shiny Server** e outra que contém o **Browser**, apesar de estar ilustrado que são duas máquinas diferentes, estas podem ser a mesma máquina se a ferramenta for executada localmente.

A ferramenta pode ser disponibilizada pela *web* através de dois métodos distintos. O primeiro método é hospedar a aplicação numa máquina Linux/Windows que contém ferramentas para disponibilizar o acesso a outros utilizadores, através da i*nternet*, isto é, com ferramentas como o Apache ou o Microsoft Information Services, caso seja Linux ou Windows, respetivamente. O segundo método é hospedar a aplicação na *cloud*, através do shinyapps.io (Rstudio, INC, 2017). Este segundo método é um serviço que permite hospedar aplicações Shiny na *web* em poucos minutos, no entanto requere uma taxa mensal ou anual e é hospedado num ambiente do qual não se tem controlo, pois é um gerido pelo RStudio.

Para ambos os métodos descritos é preciso ter o RStudio e o Shiny, no entanto para o primeiro método é preciso o Shiny Server que é um programa de *back-end*, descrito na secção 5.6.

7 Fera

Neste capítulo é descrita detalhadamente a ferramenta desenvolvida a que chamamos **FERA**, procurando refletir sobre as suas funcionalidades, as técnicas utilizadas, assim como uma avaliação geral da ferramenta e as considerações da **FERA.**

Como referido anteriormente, a **FERA** é uma aplicação Shiny, desenvolvida no RStudio, que permite extrair e visualizar regras de associação a partir de dados contidos em ficheiros XML e CSV.

A interface gráfica é composta por separadores, para visualização dos diferentes gráficos e das regras de associação construídas, e botões, para escolher os diferentes parâmetros de forma a obter regras mais interessantes de acordo com o caso em estudo, conforme ilustrado na Figura 16.

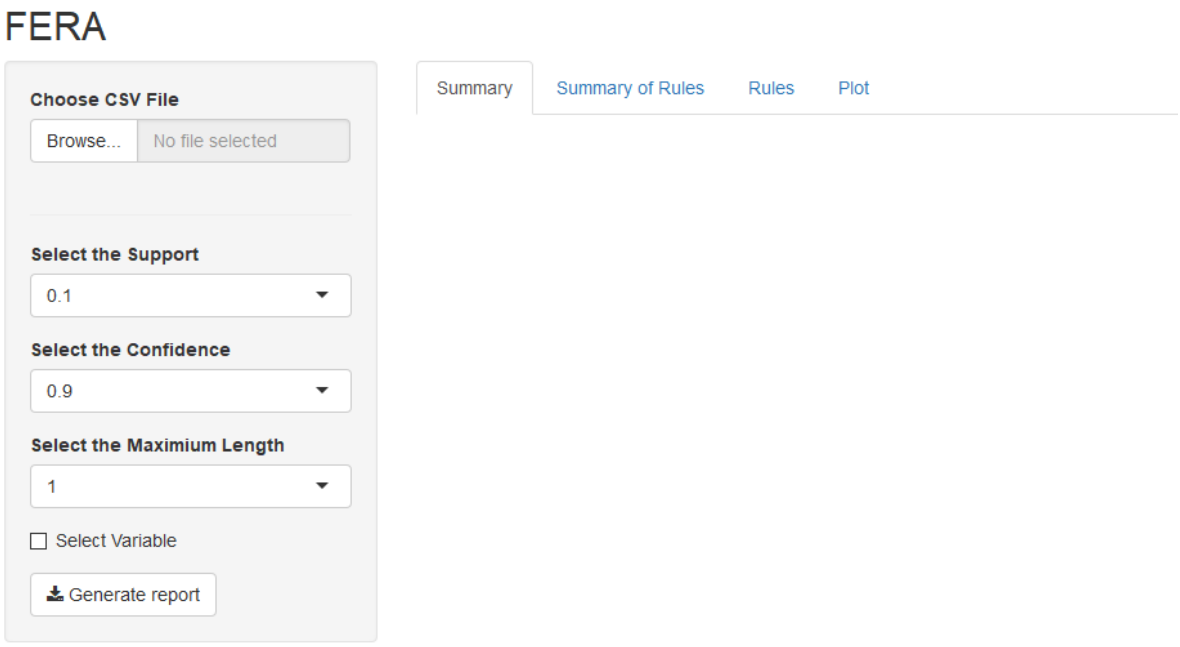

Figura 16 - Interface gráfica da FERA

Após construção das regras é possível visualizá-las a partir de um gráfico de dispersão ou numa tabela onde são enumeradas as regras geradas.

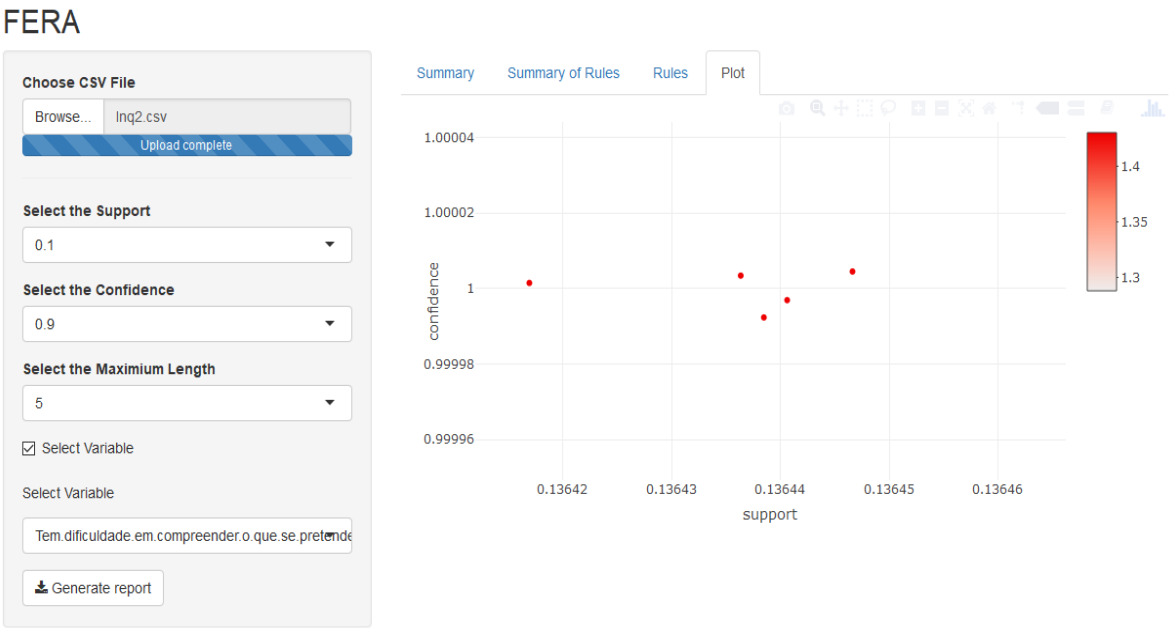

Figura 17 - Gráfico de Regras de Associação da FERA

No desenvolvimento da ferramenta houve preocupação em obter uma estrutura gráfica simples e com uma interface gráfica intuitiva para utilizador.

7.1 Funcionalidades

A **FERA** disponibiliza ao utilizador várias funcionalidades. O diagrama da figura seguinte (Figura 18) resume toda a interação que o utilizador poderá ter com a dita ferramenta, tal como já foi mencionado nos diagramas de sequência no capítulo 0.

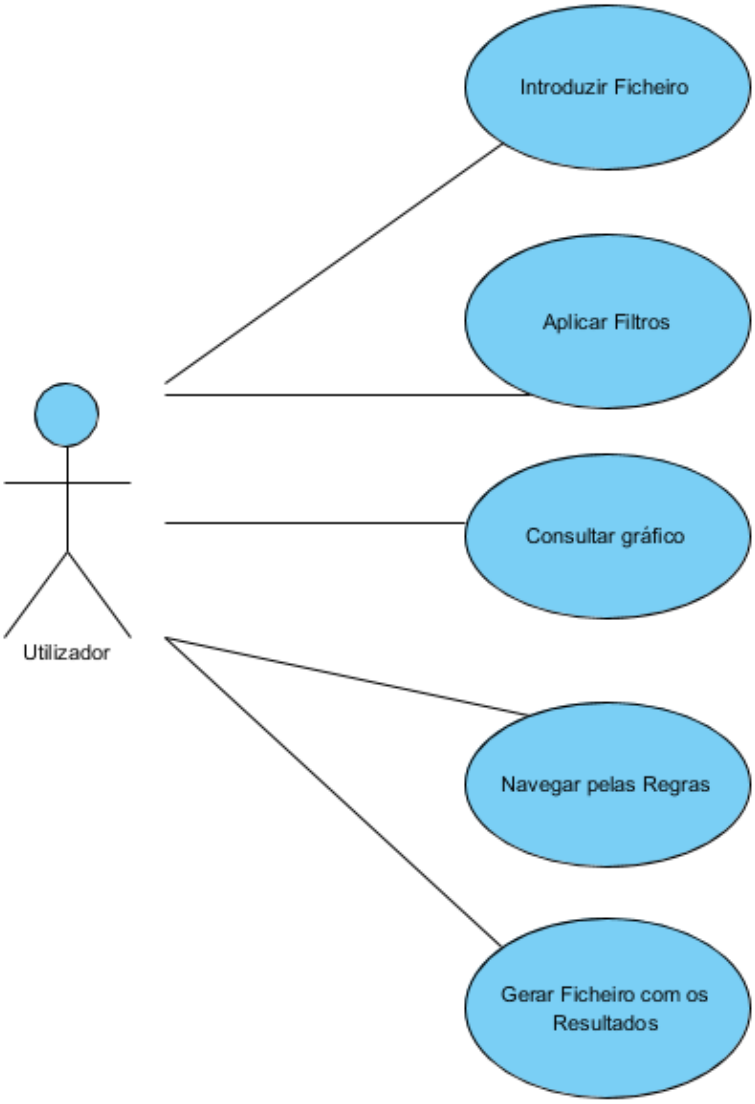

Figura 18 - Funcionalidades da FERA

7.2 Técnicas de *data mining* utilizadas

No desenvolvimento da ferramenta **FERA** foram utilizadas técnicas de pós-processamento estudadas ao longo do projeto e já descritas na secção 4.4.

Dado que as respostas dos *google forms* podem ser facilmente extraídos para uma folha de Excel e a ferramenta foi desenvolvida para uso educacional, optou-se por definir que os dados a serem introduzidos na ferramenta são provenientes de um ficheiro XML ou CSV.

As regras de associação são geradas a partir do algoritmo Apriori, contido na biblioteca 'arules' do R. Esta biblioteca foi desenvolvida por Michael Hahsler e Bettina Guren (Hahsler & Guren, 2018) e permite representar, manipular e analisar dados e padrões em *itemsets*.

Como já foi referido, aplicando algoritmos de extração de regras de associação a uma grande quantidade de dados, por norma, o número de regras geradas é grande, logo sentiu-se a necessidade de restringir o número de regras a serem geradas. Para as restringir, foram introduzidos os seguintes filtros na ferramenta:

- Suporte (Support);
- Confiança (Confidence);
- Comprimento Máximo (Maximum Length);
- Variável (Variable).

Todos os filtros contêm valores predefinidos (por omissão) no início da aplicação. No entanto, o utilizador tem liberdade de escolher os valores de cada um dos parâmetros desde que o valor exista na lista de valores disponíveis para cada atributo.

O filtro 'Comprimento Máximo' limita o número máximo de valores de um *itemset*, isto é, se contivermos a seguinte regra $\{a, b, c, d, e\}$, mas o filtro de 'Comprimento Máximo' for 4, então a regra exemplificada não será gerada pelo algoritmo.

Se selecionarmos o filtro 'Variável', então significa que todas as regras geradas terão de ter no consequente (RHS) a variável escolhida, isto é, se escolhermos a variável A, então todas as regras terão de ter no seu *itemset* essa variável.

Os valores que a 'Variável' poderá tomar são os valores do nome da coluna do ficheiro introduzido.

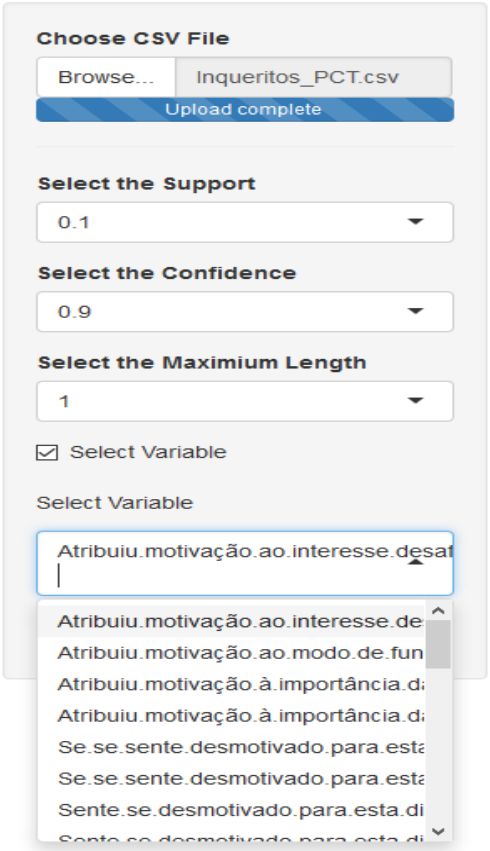

Figura 19 - Filtros da FERA

A ferramenta disponibiliza um gráfico de dispersão, que ilustra as regras de associação geradas anteriormente pela própria ferramenta, e de uma tabela que mostra as regras geradas. O gráfico é criado pela biblioteca 'plotly', biblioteca que permite gerar gráficos interativos para visualização e modificação através de um *browser*. O gráfico de dispersão permite executar as seguintes funções:

- Visualizar as regras de associação geradas;
- Navegar pelo conjunto de regras;
- Selecionar uma área do gráfico e ampliar essa área;
- Identificar cada ponto do gráfico com a regra de associação correspondente;
- Descarregar apenas o gráfico como uma imagem.

A Figura 20**Erro! A origem da referência não foi encontrada.** ilustra um gráfico de dispersão com algumas regras geradas pela **FERA**, sendo possível identificar a qual Regra de Associação corresponde a cada ponto.

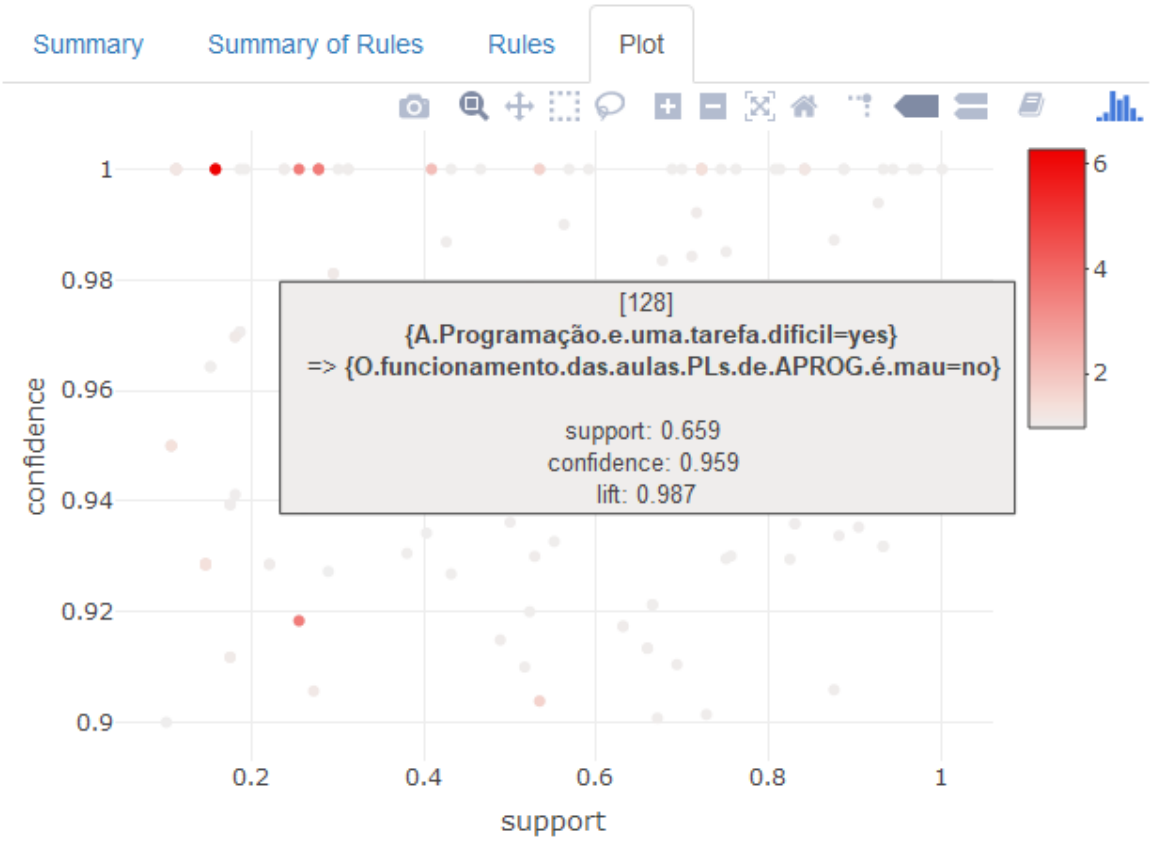

Figura 20 - Gráfico interativo da FERA

Por último, a ferramenta permite gerar um relatório dos resultados obtidos e descarregar esse ficheiro na plataforma com que o utilizador interage com a **FERA**.

O ficheiro é criado a partir da biblioteca 'rmarkdown' e que permite converter documentos R Markdown em vários formatos e foi utilizado para gerar um relatório em HTML.

O documento gerado através da **FERA** permite que todos os filtros e resultados obtidos durante o processo de geração de regras sejam incluídos no documento R Markdown.

O documento gerado é um documento interativo em formato HTML (Figura 21), visível através de um *browser* que tem a capacidade de manipular o gráfico da mesma forma que é manipulado na aplicação.

FERA Report

Diogo Vieira

```
# The `params` object is available in the document.
params$supp
```

```
## [1] "0.1"
```

```
params$conf
```

```
## [1] "0.9"
```

```
params$lift
```

```
## [1] "1"
```

```
params$maxlen
```

```
## [1] "1"
```

```
params$arules
```

```
## set of 5 rules
```

A plot of the association rules

```
plotly_arules(params$arules)
```

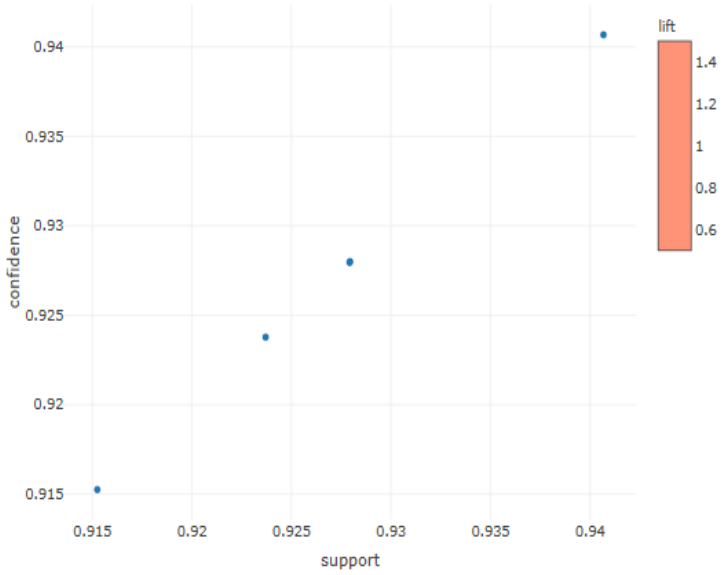

Figura 21 - Report Markdown

7.3 Avaliação da solução

Nesta seção é demonstrada a utilização da ferramenta **FERA** com dois ficheiros de dados que contém dados relativos a alunos que frequentam o Ensino Superior.

7.3.1 Questionário Motivação

O primeiro questionário contém perguntas sobre a aprendizagem no ensino superior que visa identificar as dificuldades no processo de aprendizagem da programação, nomeadamente a motivação dos alunos em disciplina de programação (ver Inquérito). Os resultados deste questionário (236 resultados), foram introduzidos na **FERA,** com o intuito de descobrir padrões sobre as questões colocadas, retirando conclusões sobre a motivação dos alunos ao longo do tempo sobre as disciplinas de programação.

Todos os casos analisados foram avaliados com um 'Comprimento Máximo' igual a 5, isto é, todos os *itemsets* retornados contém um número máximo de 5 *items.*

Começou-se por analisar todas as regras que são geradas com os seguintes filtros (valores por omissão):

- Suporte = 0.1;
- Confiança = 0.9.

No entanto, com estes valores, foram geradas 482.947 regras de associação, o que causa algum desconforto na sua análise. Foram escolhidos então dois *itens* em particular (variáveis), onde cada regra a ser gerada terá que conter um ou os dois *itens* em simultâneo.

Foram escolhidos os seguintes *itens:*

- Atribuir motivação ao interesse/desafio da matéria (Variável 1);
- Face à dificuldade da matéria sente-se desmotivado e desisto (Variável 2).

Tendo em conta estas duas variáveis, para cada uma delas, conclui-se o seguinte:

1. Variável 1

   Foram encontradas 11 regras com este *item,* no qual todas elas contém dois *itens* em comum, 'Atribui motivação à importância da matéria para a vida profissional' igual a 'Não' e 'Atribui motivação à importância da matéria para o curso' igual a 'Não', ou seja, todos os

intervenientes classificam que o facto de a matéria ser ou não importante para o curso e para a vida profissional tem relação com o interesse/desafio da matéria.

De acordo com a **FERA** e o algoritmo para gerar as regras de associação (apriori), as regras mais frequentes constam que:

- 62 dos 236 intervenientes questionados, atribuem interesse ao desafio da matéria ao facto de se sentirem motivados para ultrapassar a dificuldade das matérias e não ao modo de funcionamento das aulas.
- 50 dos intervenientes questionados, sentem-se intrinsecamente motivados para o estudo da matéria, logo classificam-na como interessante.

2. Variável 2

Com esta variável verifica-se que os intervenientes não se sentem motivados e acabam por desistir, maioritariamente, precisam de incentivos externos, gostariam de ter uma plataforma de apoio à resolução de exercícios fora das aulas e, como tal, sentem que a sua motivação diminui ao longo do semestre. É de salientar que em todas as regras geradas com os filtros definidos, todos os intervenientes que se sentem desmotivados e que acabam por desistir, nenhum deles atribui valor positivo à Variável 1.

Com base nesta análise (e para os parâmetros escolhidos), observa-se que a principal falha na motivação deve-se ao facto de os alunos perderem a motivação ao longo do semestre e não terem um auxílio para recuperar essa motivação a não ser deles próprios, e que o interesse da matéria não advém das saídas profissionais que contém ou da importância que é para o seu curso, mas sim pelas suas características.

Para uma melhor avaliação da solução seria importante aferir a satisfação dos utilizadores no sentido de avaliar se a solução proposta acrescenta valor no que diz respeito ao pós-processamento das regras, ao manuseamento da ferramenta e à qualidade de informação fornecida (por exemplo visualização das regras e gráficos apresentados). Para tal seria necessário recolher respostas a um inquérito de satisfação.

7.3.2 Questionário APROG

Este questionário contém perguntas sobre a aprendizagem na disciplina de Algoritmia e Programação (APROG) dos alunos do Instituto Superior de Engenharia do Porto (ISEP) (ver Inquérito).

Para o questionário APPROG (176 resultados), foram efetuados exatamente os mesmo testes que o questionário motivação.

Começou-se por analisar todas as regras que são geradas com os seguintes filtros (valores por omissão):

- Suporte = 0.1;
- Confiança = 0.9.

Para reduzir o número de regras geradas, somente com valores por omissão, foram escolhidos os seguintes *itens:*

- Já sabia programar antes de frequentar APROG (Variável 1);
- Preferem trabalhar sozinhos (Variável 2).
- Não têm dificuldade em compreender o que se pretende nos exercícios (Variável 3).

Tendo em conta estas duas variáveis, para cada uma delas, conclui-se o seguinte:

1. Variável 1

   Com os filtros aplicados encontrou-se, somente, uma regra com este *item,* no qual indica que 18 dos intervenientes que sabiam programar antes de frequentar a disciplina de APROG, escolheram como primeira opção o curso de Engenharia Informática, não sentem dificuldade perante a matéria lecionada e não precisa de incentivos externos para se sentir motivado.

2. Variável 2

   Para esta variável e com os presentes filtros foram encontradas 1219 regras de associação. Para o caso de estudo, apenas se avaliou duas das regras com maior suporte e com um mínimo de três itens.

   - 46 dos intervenientes que responderam que preferem trabalhar sozinhos, sentem-se motivados na disciplina lecionada, contentes com a escola de Ensino e com o funcionamento das aulas práticas (regra com valor de suporte igual 0.26 e confiança igual a 1);
   - 40 dos intervenientes que responderam que preferem trabalhar sozinhos, gostariam de obter *feedback* na correção dos exercícios e não precisam de incentivos externos para se sentirem motivados (regra com valor de suporte igual 0.23 e confiança igual a 1).

3. Variável 3

Para esta variável são geradas 1015 regras e, através da **FERA** e do seu gráfico, visualizou-se as regras com um maior *lift* (Figura 22).

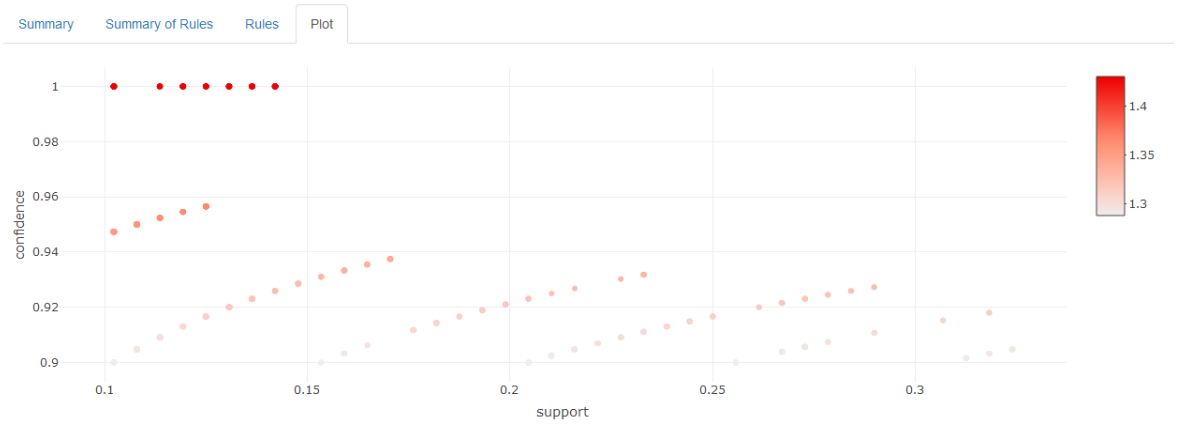

Figura 22 - Regras geradas com a variável 3

Através do gráfico, visualizou-se duas regras com o valor de confiança igual a 1 e o valor do suporte compreendido entre 0.13 e 0.15, no qual são referidas como 'Regra 1' e 'Regra 2'.

- Na Figura 23 é possível visualizar a 'Regra 1' que uma percentagem de intervenientes que responderam que não contém dificuldade na aprendizagem, não preferem trabalhar em grupo, não sentem dificuldades quanto à programação, não sentem que o funcionamento das aulas é mau e sentem que o ISEP é uma boa instituição de ensino;

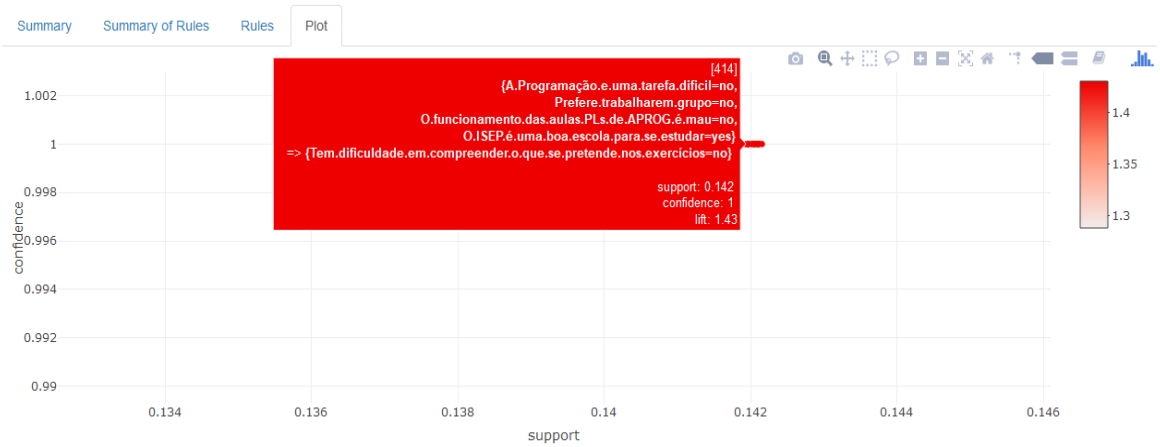

Figura 23 – Regra 1

- Na ‘Regra 2’ (Figura 24) é visível que um número de intervenientes não sentem dificuldades na programação, sentem-se motivados para a disciplina de APROG e preferem trabalhar sozinhos, logo, não contém dificuldade em compreender o que se pretende nos exercícios.

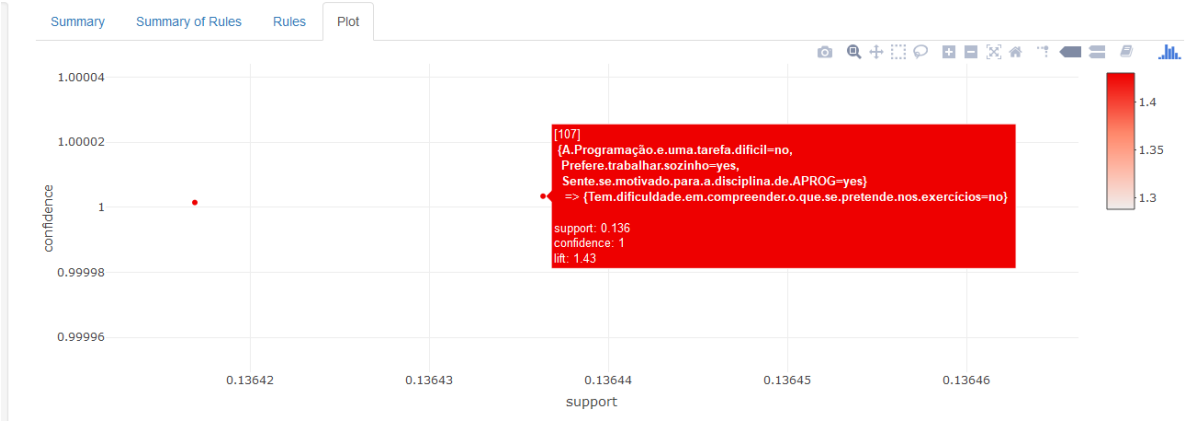

Figura 24 - Regra 2

Com base na análise das variáveis escolhidas, observa-se que, no geral, os alunos não sentem muitas dificuldades na aprendizagem desta e gostam da aprendizagem, quer seja na disciplina de APROG, quer no ISEP.

Para uma melhor avaliação da **FERA** seria importante aferir a satisfação dos utilizadores no sentido de avaliar se a solução proposta acrescenta valor no que diz respeito ao pós-processamento das regras, ao manuseamento da ferramenta e à qualidade de informação fornecida (por exemplo visualização das regras e gráficos apresentados). Para tal seria necessário recolher respostas a um inquérito de satisfação do utilizador.

7.4 Considerações

Após desenvolvida a ferramenta **FERA** faz todo o sentido apontar os pontos fortes e fracos.

Em termos globais a ferramenta desenvolvida cumpre o seu objetivo, pois permite extrair e visualizar regras de associação, logo assume-se como um contributo para o processo de extração de regras de associação de *data mining*.

A interface simples da **FERA** permite uma aprendizagem rápida, possibilitando assim, uma utilização imediata por parte dos utilizadores. A simplicidade da interface está associada ao ambiente em que o sistema foi desenvolvido: a *Internet.* Deste modo, ficam asseguradas algumas vantagens relacionadas tendo por base a não instalação do sistema, uma vez que pode ser utilizador remotamente, através de um *browser.*

Outra vantagem é o facto de se poder utilizar ficheiros provenientes do *google forms* com as respostas e analisar esses mesmos ficheiros na **FERA.**

Quantos aos aspetos negativos, no decorrer da utilização da ferramenta foi notado que, por vezes, a geração das regras de associação tende a demorar e a diminuir o desempenho da máquina se não forem aplicados filtros de modo a diminuir o número de regras a serem geradas.

No decorrer da tese foram executados alguns testes de forma a avaliar o desempenho da máquina e o tempo que demora a executar o algoritmo Apriori para gerar as regras de associação.

De notar que todos os testes foram executados num computador portátil e usado o Shiny Server para o processamento da aplicação. Na tabela seguinte é ilustrado os testes efetuados e o tempo de processamento que cada um obteve, tanto na geração das regras como na geração do gráfico.

Número de regras geradas	Tempo de processamento (segundos)	Tempo a processar o gráfico (segundos)
6.198	0.1	0.2
2.090.924	1.27	6
6.519.517	3.38	34
15.557.306	8.9	Não determinado

Tabela 5 - Testes de processamento

Por norma os testes de processamento ocorrem sem nenhum problema e confortavelmente rápidos com a exceção do último teste, onde não foi possível determinar o tempo de processamento na geração do gráfico com as regras, devido à incapacidade do processador. Com um número tão grande de regras a máquina não continha recursos suficientes para executar a ação em tempo útil.

Com base nos testes, conclui-se que a **FERA**, quando o número de regras geradas é bastante elevado, a máquina começa a deteriorar em termos de processamento, chegando mesmo a não ser capaz de a utilizar durante breves momentos.

8 Conclusões

A descoberta de regras de associação, introduzida por Agrawal, é uma técnica de *data mining,* que permite extrair conhecimento a partir de grandes volumes de dados. A interpretação e análise de regras de associação, geradas a partir de pequenos volumes de dados não apresenta qualquer dificuldade. O mesmo não se pode dizer quando o volume de dados é de tal ordem elevada.

Pelo motivo descrito acima e pela necessidade de perceber, as mudanças de cursos no Ensino Superior pelos alunos, a falta de motivação destes e de forma a prever o seu comportamento na educação, sentiu-se a necessidade de desenvolver uma ferramenta com o intuito de extrair regras de associação, compatível com o *google forms,* para assim ajudar a perceber as causas destes problemas através de inquéritos feitos aos alunos.

Esta dissertação debruçou-se sobre o desenvolvimento de uma ferramenta que pretende facilitar a interpretação e análise de um grande número de regras de associação, geradas pela própria ferramenta.

Através deste trabalho, foi produzida a **FERA** com o intuito de criar valor, não só para as organizações de Ensino Superior assim como para os seus estudantes, permitindo tirar conclusões a partir de casos de estudo.

A **FERA** foi desenvolvida em linguagem R, que fornece uma variedade de técnicas estatísticas e gráficas que podem ser facilmente utilizadas através de uma interface visual, o RStudio.

8.1 Objetivos alcançados

O objetivo principal do presente trabalho foi efetuar um estudo sobre as técnicas de *data mining*, aprendendo os conceitos e a partir desses conceitos, desenvolver uma ferramenta como a **FERA.**

Neste trabalho, foram completados todos os casos de uso propostos durante o desenvolvimento da ferramenta **FERA**.

Apesar de os objetivos terem sido alcançados e de a ferramenta desenvolvida cumprir as funções requisitadas, poderiam ser melhoradas algumas questões relativamente à ferramenta, introduzindo mais funcionalidades a esta.

8.2 Limitações

A limitação que mais afetou o desenvolvimento da **FERA**, foi a demora que existiu na aprendizagem dos conceitos de *data mining,* necessários para a desenvolver.

Para além disso, a escolha pelo *software* optado, RStudio e Shiny, também levou a algumas limitações de desenvolvimento, pela falta de conhecimento à priori destas metodologias.

Apesar das limitações não terem sido completamente ultrapassadas, uma vez que poderia ser implementada melhorias na ferramenta, foi desenvolvida uma ferramenta que cumpre os seus objetivos.

8.3 Trabalho futuro

Relativamente às melhorias que podem ser implementadas na **FERA**, apresenta-se com maior destaque a implementação de novos algoritmos para a extração de regras de associação, dando ao utilizador a possibilidade de escolher o algoritmo pelo qual quer gerar as regras. No entanto, este ponto poderia criar alguma confusão ao utilizador, caso este utilizador não tenha nenhum conhecimento dos algoritmos de regras de associação.

A aplicação **FERA** poderia conter mais gráficos que ilustrassem as regras de associação geradas.

Ao utilizar a ferramenta para a sua avaliação, verificou-se que, quando o número de regras de associação era substancialmente grande, o desempenho da máquina tornava-se lento. Esta questão poderia ser ultrapassada através da disponibilização *web* da **FERA** num servidor apropriado e, se essa medida não fosse suficiente, aumentar o número de filtros ou diminuir os valores tabelados dos filtros existentes.

Implementar a **FERA** num ambiente *web* seria o ideal para a execução desta, devido à necessidade de processamento que esta requere quando o número de regras de associação é elevado.

Por último, outra melhoria que poderia ser efetuada seria existir a possibilidade de para além de carregar ficheiros com dados para gerar regras de associação, carregar também ficheiros com regras de associação, permitindo navegar pelo conjunto de regras de associação na **FERA** através dos seus gráficos.

8.4 Apreciação final

Fazendo um balanço de todo o trabalho desenvolvido, é possível concluir que o mesmo acrescentou muito valor a nível profissional e pessoal.

A nível pessoal, permitiu a aprendizagem de novos conceitos, como *data mining,* EDM e Shiny. Permitiu também aumentar as competências ao nível do trabalho e a um planeamento entre a vida profissional e escolar, o que levou a um aumento de responsabilidade e compromisso pessoal.

A nível profissional permitiu aprender novas tecnologias, podendo abrir portas futuras no ramo de *data mining*.

Relativamente ao projeto proposto considera-se que o trabalho desenvolvido teve um impacto positivo, pois, tendo em conta que foi um projeto criado de raiz, foi desenvolvida uma boa versão da aplicação, pensada e desenvolvida considerando os pontos propostos e de potencial evolução.

9 Referências

Agathe, M. & Yacef, K., 2004. TADA-Ed, a tool to visualize and mine students. *Proceedings of International Conference on Computers in Education.*

Agrawal, R., Swami, A. & Imielinski, T., 1993. *Mining Association Rules between Sets of Items in Large Databases,* s.l.: s.n.

Basu, S., Mooney, R., Pasupuleti, K. & Ghosh, J., 2001. *Evaluating the Novelty of Text-Mined Rules using Lexical Knowledge ,* s.l.: s.n.

Domingues, M. & Rezende, S., 2005. *Pós-processamento de Regras de Associação usando Taxonomias,* s.l.: s.n.

Eeles, P., 2004. *FURPS+.* [Online] Available at: https://www.ibm.com/developerworks/rational/library/3975.html [Acedido em 2018].

Engrácia, P. & Baptista, J., 2018. Percursos no ensino superior: situação após quatro anos dos alunos inscritos em licenciaturas de três anos. Março, p. 30.

Fielding, R. & Irvine, U., 1999. *Hypertext Transfer Protocol,* s.l.: s.n.

Fu, Y., s.d. *Data Mining: Tasks, Techniques, and Applications,* s.l.: s.n.

Geng, L. & Hamilton, J., 2006. Interestingness Measures for Data Mining: A Survey. *ACM Computing Survey,* Volume 38.

Gomes, E., Tavares, P. & Henriques, P., 2018. *Studying Programming Students Motivation using Association Rules.* s.l., Science and Technology Publications, Lda.

Gremberghe, I. v., 2016. *Introduction to Markdown,* s.l.: s.n.

Hahsler, M. & Guren, B., 2018. *Mining Association Rules and Frequent Itemsets.* [Online] Available at: https://cran.r-project.org/web/packages/arules/arules.pdf [Acedido em 2018].

Hornik, K., 2017. *Frequently Asked Questions on R.* [Online] Available at: https://cran.r-project.org/doc/FAQ/R-FAQ.html [Acedido em 2018].

Industrial Research Institute, Inc, 2001. Providingclarity and a common language to the "fuzzy front end". Volume 44.

Jacob, J., Jha, K., Kotak, P. & Puthran, S., 2015. *Education Data Mining Techniques and their applications,* s.l.: Internation Conference on Green Computing and Internet of Things.

Kamber, J. H. a. M., 2006. *Data Mining Concept and Techniques,* s.l.: s.n.

Kim, J. H., 2014. *Apriori Algorithm,* s.l.: s.n.

Koen, P., 2004. Understanding the Front End. *A Common Language and Structured Picture*, 25 Maio.

Kularbphettong, K. & Tongsiri, C., 2012. Mining Educational Data to Analyze the Student Motivation Behavior. *International Journal of Information and Communication Engineering,* Volume 6.

Kumbhare, T. & Chobe, S., 2014. An Overview of Association Rule Mining Algorithms. *Internation Journal of Computer Science and Information Techiniques,* Volume 5.

Lenca, P., Vaillant, B., Meyer, P. & Lallich, S., s.d. *Association rule interestingness measures: experimental and theoretical studies,* s.l.: s.n.

Maindonald, J., 2008. *Using R for Data Analysis and Graphics,* s.l.: s.n.

Maksood, F. & Achuthan, G., 2006. Analysis of Data Mining Techniques nad its Applcations. *Internation Journal of Computer Applications,* Abril.Volume 140.

Merceron, A. & Yacef, K., 2005. *Educational Data Mining: a Case Study,* Sydney, Australia: DBLP.

Ozer, P., 2008. *Data Mining Algorithms for Classification,* s.l.: Radboud University Nijmegen.

Pujari, P. & Gupta, J., 2012. Exploiting data mining techniques for improving the efficiency of time series data using SPSS-Clementine. *International Refereed Research Journal,* 3(2).

Qt Group, 1995. *Qt Framework.* [Online] Available at: https://www.qt.io/

Rocha, R., 2006. Ambiente Web Extensível para Pós-Processamento de Regras de Associação. Julho.

Romero, C., Ventura, S., Pechenizkiy, M. & Baker, R. S. J. d., 2007. *Handbook of Educational Data Mining,* s.l.: Chapman & Hall/CRC.

RStudio, Inc, 2011. *RStudio, new open-source IDE for R.* [Online] Available at: https://blog.rstudio.com/2011/02/28/rstudio-new-open-source-ide-for-r/ [Acedido em 2018].

RStudio, Inc, 2015. *About Rstudio.* [Online] Available at: https://www.rstudio.com/about/ [Acedido em 2018].

Rstudio, INC, 2017. *Share your Shiny Applications Online.* [Online] Available at: https://www.shinyapps.io/ [Acedido em 2018].

Rstudio, Inc, 2017. *Shiny.* [Online] Available at: http://shiny.rstudio.com/ [Acedido em 2018].

Rustemi, A. & Halili, F., 2016. Predictive Modeling: Data Mining Regression Techinique Applied in a Prototype. *Internation Journal of Computer Science and Mobile Computing,* 5(8), pp. 207-215.

Saa, A. A., 2016. Education Data Mining & Students Performance Prediciton. *Internation Journal of Advanced Computer Science and Applicaitons,* Volume 7.

Sayad, S., 2006. *Data Mining - Classification & Prediction.* [Online] Available at: https://www.tutorialspoint.com/data_mining/dm_classification_prediction.htm

Sayad, S., 2006. *Data Mining - Cluster Analysis.* [Online] Available at: https://www.tutorialspoint.com/data_mining/dm_cluster_analysis.htm

Sayad, S., 2008. *Association Rules.* [Online] Available at: https://www.saedsayad.com/association_rules.htm [Acedido em July 2018].

Sultan, A., 2004. *Subjective Measures and their Role in Data Mining Process,* s.l.: International Conference on Cognitive Systems.

Verhein, F., 2008. *Frequent Pattern Growth (FP-Growth) Algorithm,* s.l.: s.n.

Vlatko Nikolovski, R. S. I. C. I. M., 2014. *Education Data Mining: Case Study for Predicting Sudent Dropout in Higher Education,* s.l.: s.n.

Wilson, M., s.d. *What is IIS? A Basic Tutorial of the Windows Web Server.* [Online] Available at: https://searchwindowsserver.techtarget.com/definition/IIS [Acedido em 2018].

10 Anexos

10.1 Inquérito Motivação

Agradeço que preencha este formulário anónimo no contexto de um estudo sobre as dificuldades no processo de ensino/aprendizagem da Programação.

Responda com o máximo de sinceridade e sem hesitações.

1 - Se se sente motivado para esta disciplina, a que fatores atribui essa motivação: ☐ Ao interesse/desafio da matéria ☐ Ao modo de funcionamento das aulas ☐ À importância da matéria para o curso ☐ À importância da matéria para a vida profissional	**2** - Se se sente desmotivado para esta disciplina, a que fatores atribui essa desmotivação: ☐ Dificuldade da matéria ☐ Ao modo de funcionamento das aulas ☐ Não reconhecer a importância da matéria ☐ Falta de acompanhamento por falta de tempo ou por falta de bases

3 - Sente que a sua motivação diminuiu ao longo do semestre? ☐ Sim ☐ Não	**4** - Face à dificuldade das matérias da disciplina: ☐ Sinto-me motivado para as ultrapassar ☐ Sinto-me desmotivado e desisto

5 - Principal razão pela qual estudo é: ☐ Adquirir novos conhecimentos ☐ Enfrentar desafios ☐ Tirar boas notas ☐ Agradar aos outros (professores, família ou colegas)	**6** - Tenho dificuldade em compreender exatamente o que se pretende perguntar nos exercícios e questões das provas. ☐ Sim ☐ Não

7 - Sei como me valorizar nas provas e trabalhos para avaliação (sei como usar o tempo/esforço). ☐ Sim ☐ Não	**8** - Sinto-me desmotivado porque não tenho um feedback da correção dos exercícios. ☐ Sim ☐ Não

9 - Sinto-me intrinsecamente motivado para o estudo desta matéria ☐ ou preciso de incentivos externos ☐.

10 - Sou adepto do uso de plataformas digitais de ensino à distância (eLearning), tipo Moodle ou Blackboard. ☐ Sim ☐ Não	**11** - Gostaria de poder usar uma plataforma de apoio à resolução de exercícios fora da aula que desse feedback, mostrasse soluções ou guiasse a resolução. ☐ ¢Sim ☐ Não

Agradeço que preencha este formulário anónimo no contexto de um estudo sobre as dificuldades no processo de ensino/aprendizagem da Programação.

Responda com o máximo de sinceridade e sem hesitações.

1 - Se se sente motivado para esta disciplina, a que fatores atribui essa motivação: ☐ Ao interesse/desafio da matéria ☐ Ao modo de funcionamento das aulas ☐ À importância da matéria para o curso ☐ À importância da matéria para a vida profissional	**2** - Se se sente desmotivado para esta disciplina, a que fatores atribui essa desmotivação: ☐ Dificuldade da matéria ☐ Ao modo de funcionamento das aulas ☐ Não reconhecer a importância da matéria ☐ Falta de acompanhamento por falta de tempo ou por falta de bases

3 - Sente que a sua motivação diminuiu ao longo do semestre? ☐ Sim ☐ Não	**4** - Face à dificuldade das matérias da disciplina: ☐ Sinto-me motivado para as ultrapassar ☐ Sinto-me desmotivado e desisto

5 - Principal razão pela qual estudo é: ☐ Adquirir novos conhecimentos ☐ Enfrentar desafios ☐ Tirar boas notas ☐ Agradar aos outros (professores, família ou colegas)	**6** - Tenho dificuldade em compreender exatamente o que se pretende perguntar nos exercícios e questões das provas. ☐ Sim ☐ Não

7 - Sei como me valorizar nas provas e trabalhos para avaliação (sei como usar o tempo/esforço). ☐ Sim ☐ Não	**8** - Sinto-me desmotivado porque não tenho um feedback da correção dos exercícios. ☐ Sim ☐ Não

9 - Sinto-me intrinsecamente motivado para o estudo desta matéria ☐ ou preciso de incentivos externos ☐.

10 - Sou adepto do uso de plataformas digitais de ensino à distância (eLearning), tipo Moodle ou Blackboard. ☐ Sim ☐ Não	**11** - Gostaria de poder usar uma plataforma de apoio à resolução de exercícios fora da aula que desse feedback, mostrasse soluções ou guiasse a resolução. ☐ Sim ☐ Não

10.2 Inquérito APROG

Agradeço que preencha este formulário anónimo no contexto de um estudo sobre as dificuldades no processo de ensino/aprendizagem da Programação. Deve responder com o máximo de sinceridade e sem hesitações

1 – Género ☐ F ☐ M 2 – Idade _____ 3 – Ponderou não frequentar o ensino superior?	11 – Tem dificuldade em compreender o que se pretende nos exercícios? ☐ Sim ☐ Não

☐ Sim ☐ Não 4 – Este curso foi a sua primeira opção? ☐ Sim ☐ Não 5 – Qual a sua média de entrada no ensino superior? ☐ 6 – Já sabia programar antes de frequentar APROG? ☐ Sim ☐ Não 7 – Já frequentou APROG numa edição anterior? ☐ Sim ☐ Não 8 – A Programação de computadores é uma tarefa difícil? ☐ Sim ☐ Não 9 –Em trabalhos práticos prefere trabalhar: ☐ Sozinho ☐ Em grupo 10 – Sente-se motivado para a disciplina de APROG? ☐ Sim ☐ Não	12 – Gostaria de ter feedback da correção dos exercícios? ☐ Sim ☐ Não 13 – Precisa de incentivos externos para se sentir motivado para o estudo desta matéria? ☐ Sim ☐ Não 14 – Gostaria de poder usar uma plataforma de apoio à resolução de exercícios fora da aula? ☐ Sim ☐ Não 15 – De modo geral acha que funcionamento das aulas PLs de APROG é: ☐ Bom ☐ Razoável ☐ Mau 16 – Quanto ao *layout* (disposição do mobiliário) dos laboratórios acha: ☐ Boa ☐ Razoável ☐ Má 17 – O ISEP é uma boa escola para se estudar? ☐ Sim ☐ Não