Mineração de dados para geração de árvore de decisão: aplicação em vendas de varejo

Aluno: Lucas Dalcol Pereira

Orientador: Mauro Marcelo Mattos

FURB
UNIVERSIDADE DE BLUMENAU

Roteiro

- Introdução;
- Objetivos;
- Fundamentação teórica;
- Requisitos funcionais e não funcionais;
- Especificação;
- Implementação;
- Resultados;
- Conclusões.

Introdução

- O volume de dados armazenados atualmente é gigantesco e continua crescendo rapidamente pelo mundo todo (REZENDE, 2003, p. 307).

- Devido a tamanha quantidade de dados, muita informação e conhecimento úteis podem estar sendo desperdiçados (REZENDE, 2003, p.307).

Introdução

- As estratégias para obtenção de ganho de competitividade entre as empresas devem ser baseadas em informações concretas, visando a minimização de erros para a tomada de decisões por parte dos gestores (DANTAS et al., 2008, p. 1).

Introdução

- Existem ferramentas que auxiliam na coleta de dados para estratificação e classificação de informações, como a ferramenta Waikato Environment for Knowledge Analysis (Weka).

- O termo Knowledge Discovery in Databases (KDD) foi formalizado em 1989 em referência ao conceito de identificar conhecimento a partir de base de dados (MACEDO; MATOS, 2010).

Introdução

- Este trabalho propõe o desenvolvimento de um protótipo de software de mineração, o qual, aplicando técnicas de KDD, utilize o algoritmo para geração de árvore de decisão da ferramenta de mineração de dados Weka.
- A aplicação executará sobre um extrato de uma base de dados coletados de uma empresa de vendas de varejo. A expectativa é que a árvore de decisões obtida possa, de alguma forma, contribuir para a tomada de decisões na empresa.

Objetivos

O objetivo geral deste trabalho é desenvolver um protótipo de software que realize a mineração de dados em informações de vendas de varejo para geração de árvore de decisão.

Objetivos

Os objetivos específicos são:

a) modelar na prática todas as etapas do processo de KDD;
b) disponibilizar uma aplicação que permita a visualização da árvore de decisão produzida;
c) validar o modelo produzido a partir de uma base de dados de exemplo.

Fundamentação Teórica

Ferramenta Weka:

- Desenvolvida pela Universidade de Waikato, Nova Zelândia;
- Linguagem Java;
- Open Source;
- Ferramenta de KDD;
- 56 algoritmos de classificação;
- Sistemas de aprendizagem.

Fundamentação Teórica

Arquivo ARFF - Attribute-Relation File Format:

```
@RELATION iris

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

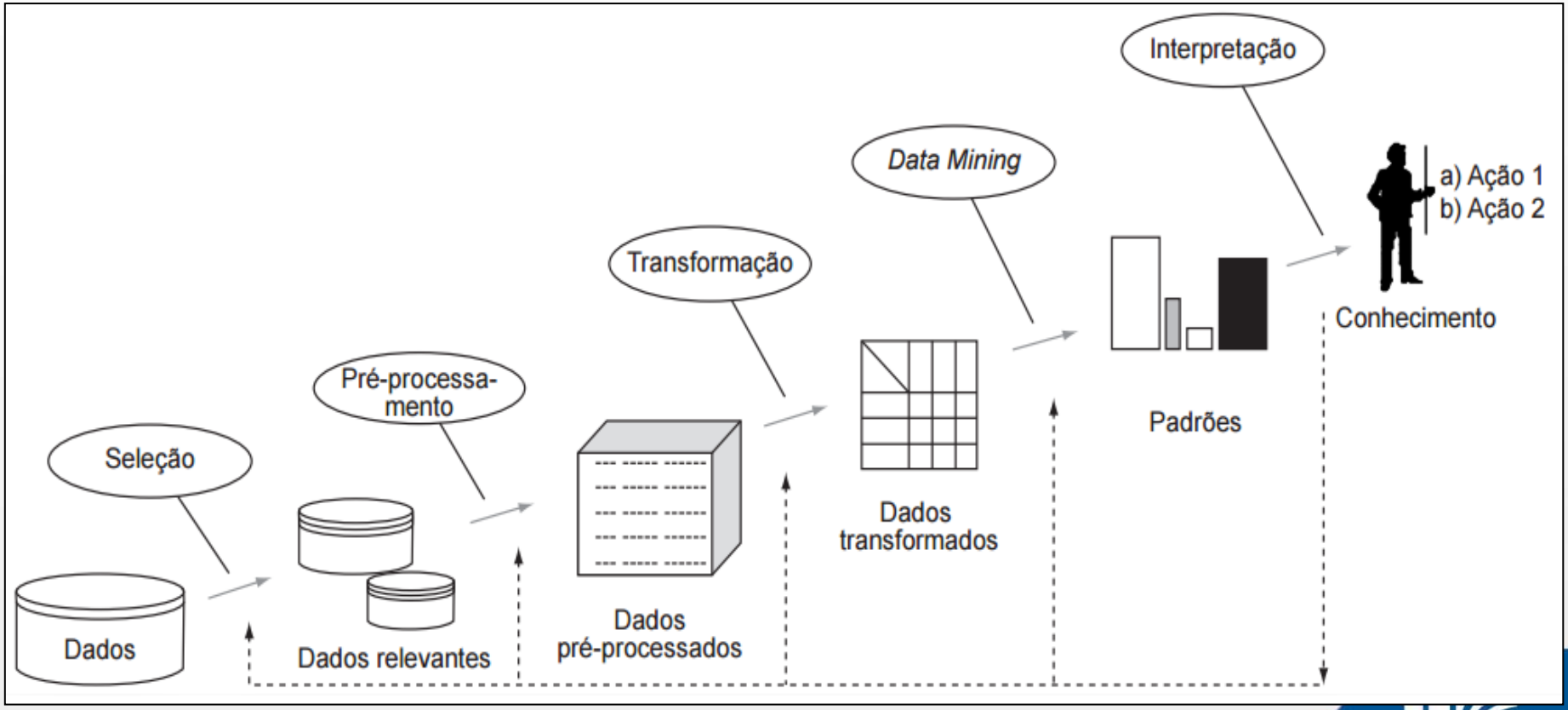
Fundamentação Teórica

Algoritmo J48:

- Baseado no C4.5;
- Java;
- Grande utilização por especialistas em mineração de dados;
- Apresenta o melhor resultado na geração de árvore de decisão quando aplicado em dados de treinamento (LIBRELOTTO; MOZZAQUATRO, 2013).
- Ganho de informação e relação de ganho padrão que divide a informação adquirida pela informação fornecida pelos resultados do teste (WU et al., 2007).

Fundamentação Teórica

Etapas do KDD:



Trabalhos Correlatos

- REDECA (Juste, 2013):
  - Aplicação de mineração de dados utilizando a ferramenta Weka.
- MinerAll (Gerosa, 2011):
  - Aplicação de mineração de dados em arquivos de repositório de software livre.
- Análise dos algoritmos de mineração J48 e Apriori (Librelotto e Mozzaquatro, 2013):
  - Aplicação de mineração de dados em arquivo ARFF e realizada a comparação entre os algoritmos.

Trabalhos Correlatos

REDECA (Juste, 2013):

- Coleta de informações sobre atendimentos de crianças e adolescentes;
- Objetivo é auxiliar no planejamento estratégico;
- Utiliza o processo de KDD;
- Utiliza o arquivo ARFF;
- Utiliza o algoritmo J48;
- Geração de árvore de decisão;
- Tentativa de mineração diretamente na base de dados.

Trabalhos Correlatos

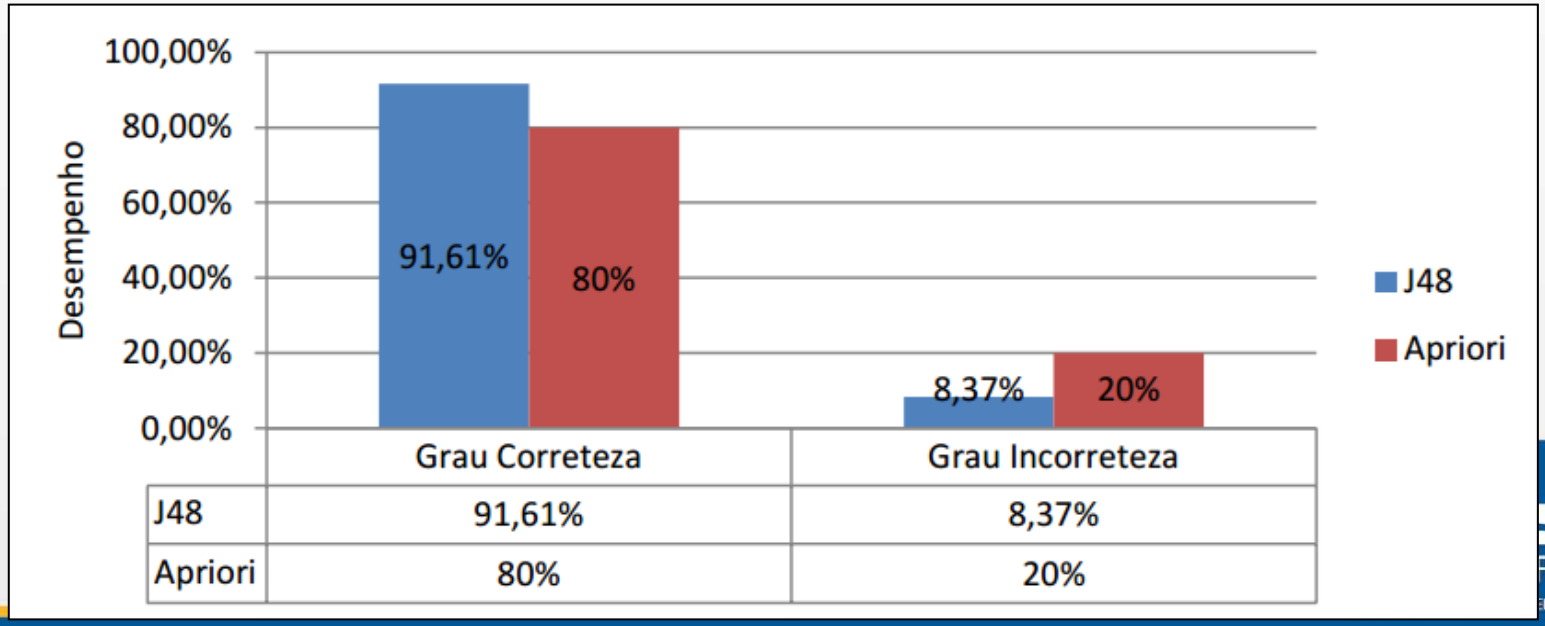
MinerAll (Gerosa, 2011):

- Mineração de dados em arquivos de código fonte;
- Objetivo identificar quais usuários são mais ativos e as dependências entre os artefatos;
- Utiliza mineração em arquivos;
- Geração de árvore de decisão;
- Geração de gráficos para apresentação dos resultados.

Trabalhos Correlatos

Análise dos algoritmos de mineração J48 e Apriori (Librelotto e Mozzaquatro, 2013):

- Coleta de dados e mineração sobre informações de perfis de indicadores de saúde (IMC, Pressão sistólica e diastólica, circunferência).



	Grau Correteza	Grau Incorreteza
J48	91,61%	8,37%
Apriori	80%	20%

Requisitos Funcionais

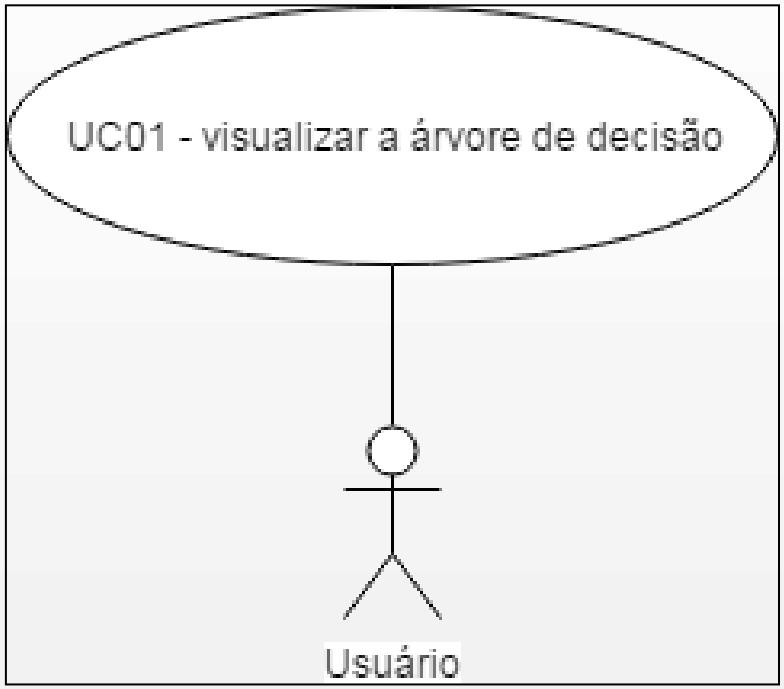
- RF01: permitir ao usuário acessar a página de introdução do software;
- RF02: permitir ao usuário gerar a árvore de decisão;
- RF03: realizar a carga do arquivo com dados classificados para mineração;
- RF04: realizar a aplicação do algoritmo de mineração de dados J48;
- RF05: realizar a apresentação da árvore de decisão gerada pelo algoritmo.

Requisitos Não Funcionais

- RNF01: as telas devem ser implementadas na linguagem de programação JavaScript;
- RNF02: o *webservice* deve ser implementado utilizando a linguagem de programação Java;
- RNF03: utilizar o algoritmo J48 da biblioteca Weka.

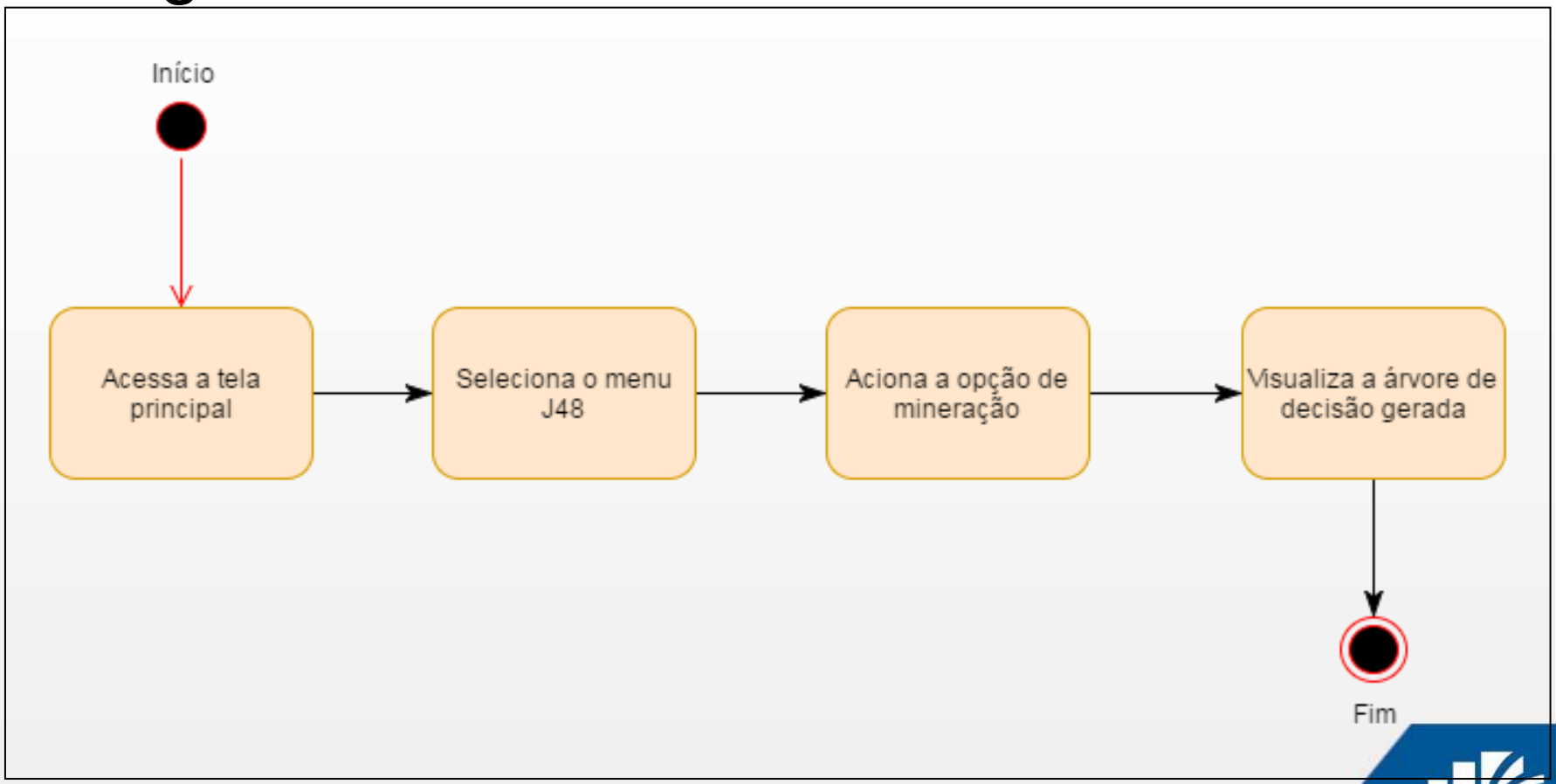
Especificação

Diagrama de Caso de Uso:



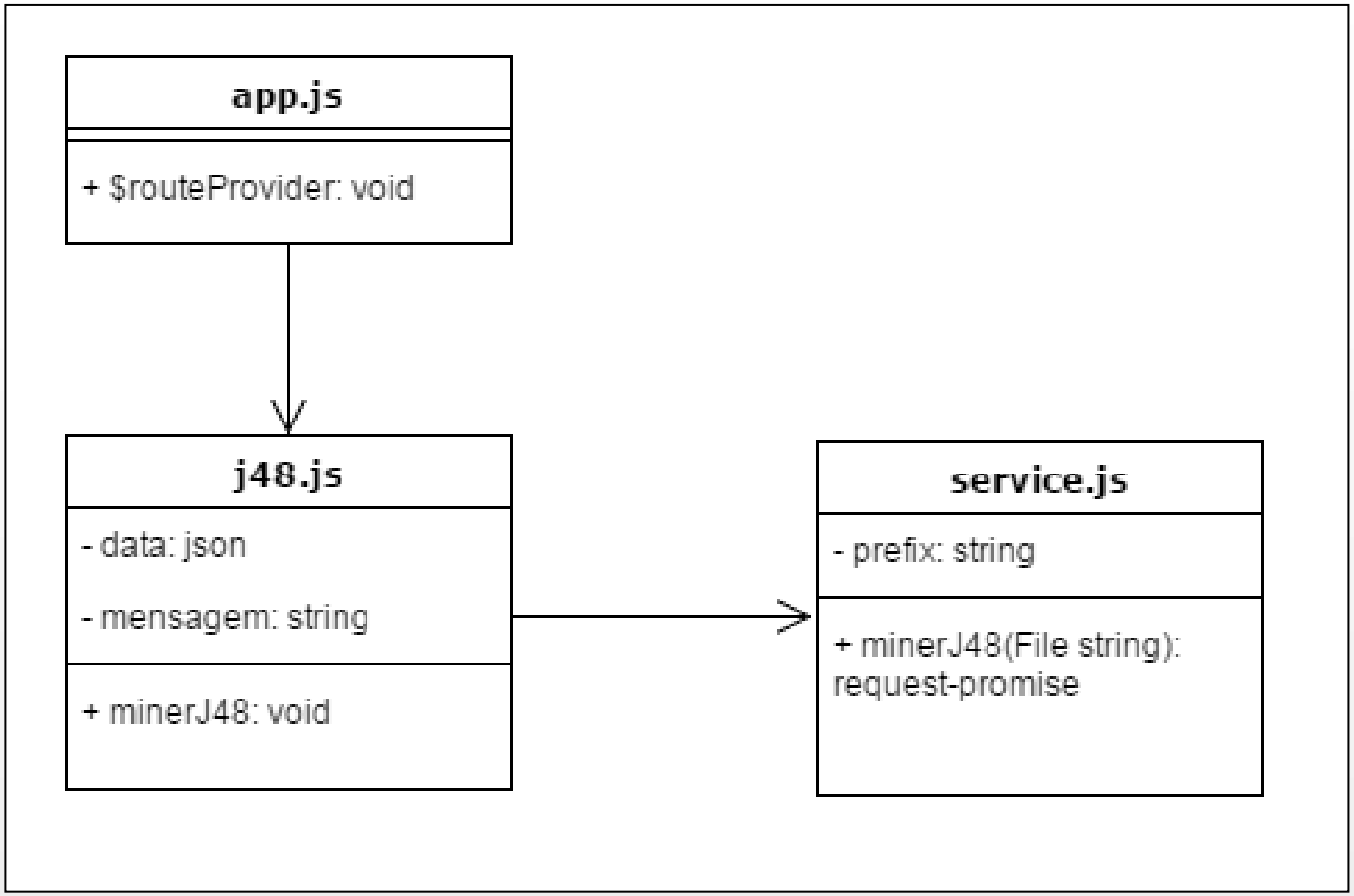
Especificação

Diagrama de atividades:



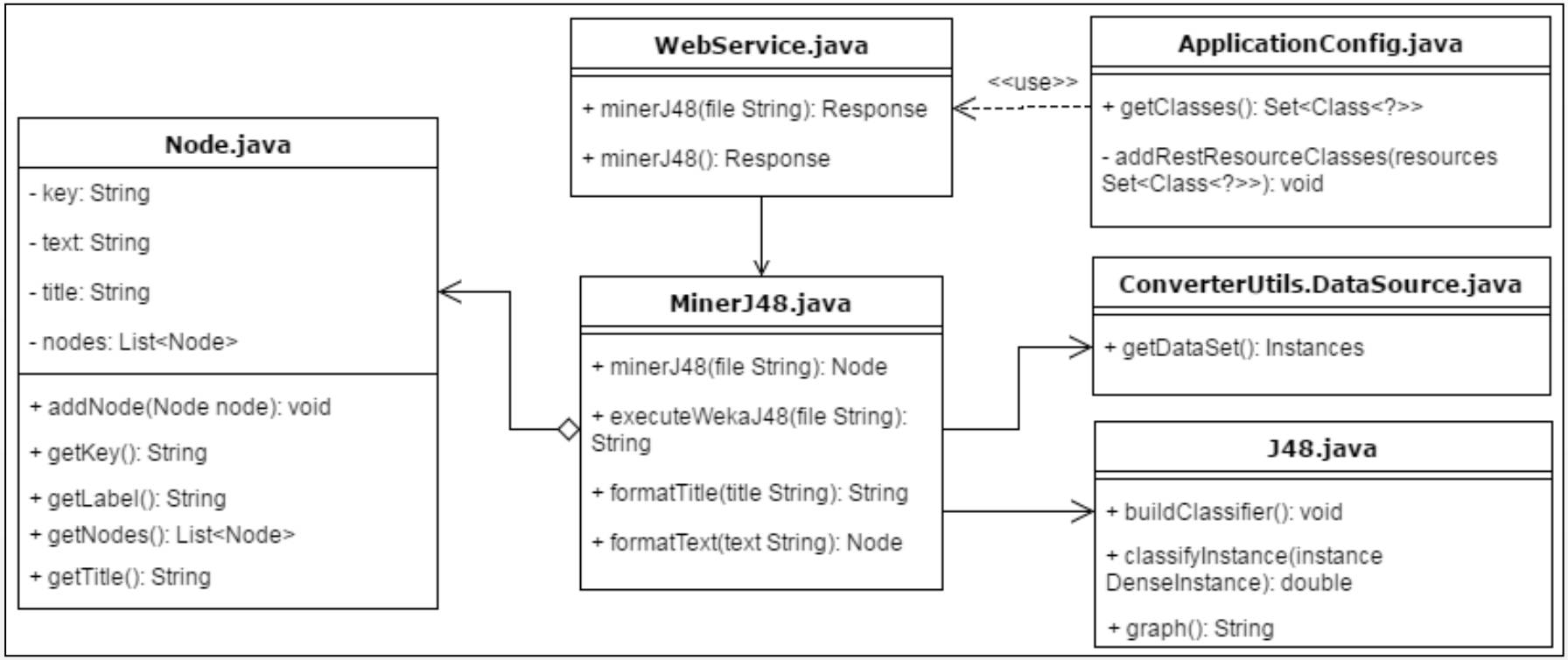
Especificação

Diagrama de classes – Front-end:



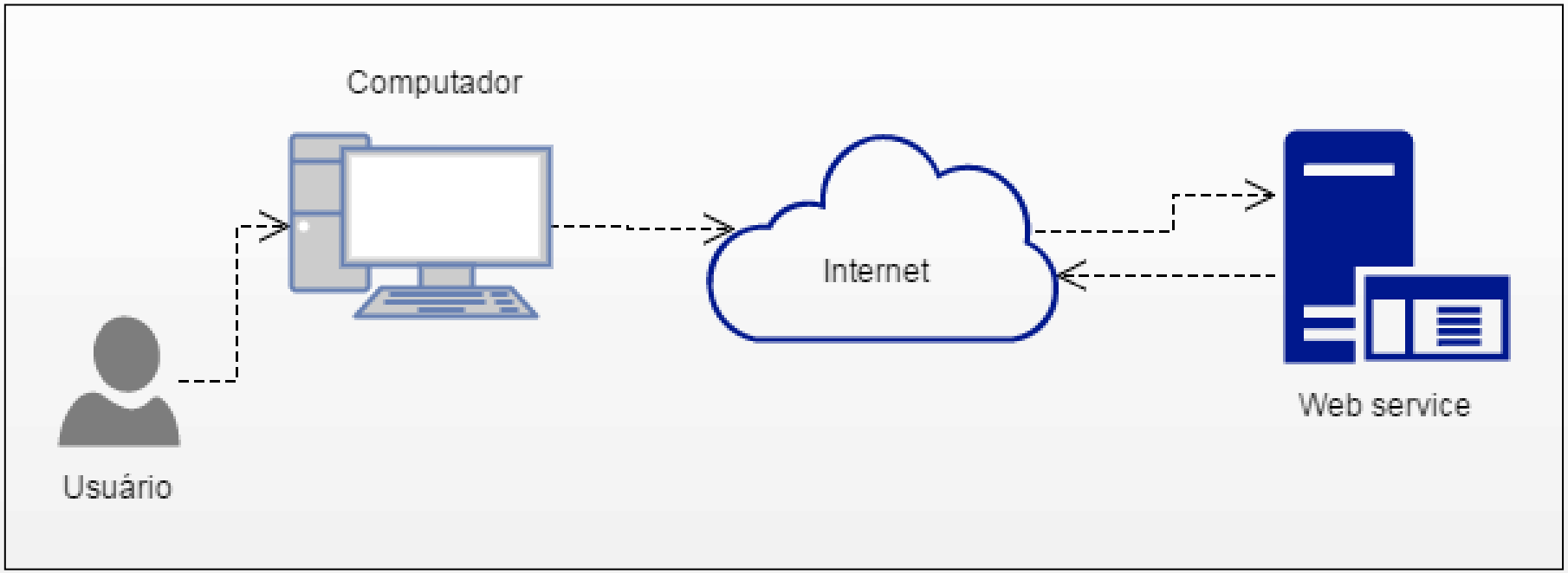
Especificação

Diagrama de classes – Back-end:



Especificação

Diagrama de deployment:



Implementação

Coleta de dados de vendas de varejo:

Filial	Data	Hora	Atendimentos	Fluxo	Fluxo vitrine
Shopping A	02/01/2017	10	7	10	172
Shopping A	02/01/2017	11	4	45	71
Shopping A	02/01/2017	12	0	20	141
Shopping A	02/01/2017	13	8	33	128
Shopping A	02/01/2017	14	5	43	84
Shopping A	02/01/2017	15	9	81	236
Shopping A	02/01/2017	16	8	77	540
Shopping A	02/01/2017	17	16	153	717
Shopping B	03/01/2017	11	0	3	25
Shopping B	03/01/2017	12	1	3	67
Shopping B	03/01/2017	13	0	11	103
Shopping B	03/01/2017	14	3	25	92
Shopping B	03/01/2017	15	0	18	103
Shopping B	03/01/2017	16	2	14	97
Shopping B	03/01/2017	17	2	21	165
Shopping B	03/01/2017	18	1	7	86

Implementação

Coleta de dados metereológicos:

- Os valores foram obtidos do Banco de Dados Metereológicos para Ensino e Pesquisa (BDMEP), que é populado com dados do Instituto Nacional de Metereologia (INMET);
- Obtidos os valores de 2 meses, janeiro e fevereiro da cidade correspondente à cada Shopping do extrato da base de vendas de varejo.

Implementação

Coleta de dados metereológicos:

```
--------------------
BDMEP - INMET
--------------------
Estação            : FLORIANOPOLIS - SC (OMM: 83897)
Latitude  (graus) : -27.58
Longitude (graus) : -48.56
Altitude  (metros): 1.84
Estação Operante
Inicio de operação: 01/12/1921
Periodo solicitado dos dados: 01/01/2017 a 01/02/2017
Os dados listados abaixo são os que encontram-se digitados no BDMEP
Hora em UTC
--------------------
Obs.: Os dados aparecem separados por ; (ponto e vírgula) no formato txt.
      Para o formato planilha XLS, siga as instruções
--------------------
Estacao;Data;Hora;UmidadeRelativa;PressaoAtmEstacao;VelocidadeVentoNebulosidade;
83897;01/01/2017;0000;90;1008.8;1;10;
83897;01/01/2017;1200;78;1007.4;2;8;
83897;01/01/2017;1800;72;1007.5;4;8;
83897;02/01/2017;0000;92;1009.2;0;3;
83897;02/01/2017;1200;68;1009.5;0;4;
83897;02/01/2017;1800;73;1010.2;4;8;
83897;03/01/2017;0000;81;1011.5;4;8;
83897;03/01/2017;1200;80;1013.4;1;6;
83897;03/01/2017;1800;62;1009.9;2;7;
83897;04/01/2017;0000;80;1010;3;10;
83897;04/01/2017;1200;72;1009.2;1;7;
83897;04/01/2017;1800;70;1006.8;4;3;
```

Implementação

Geração de dados para mineração:

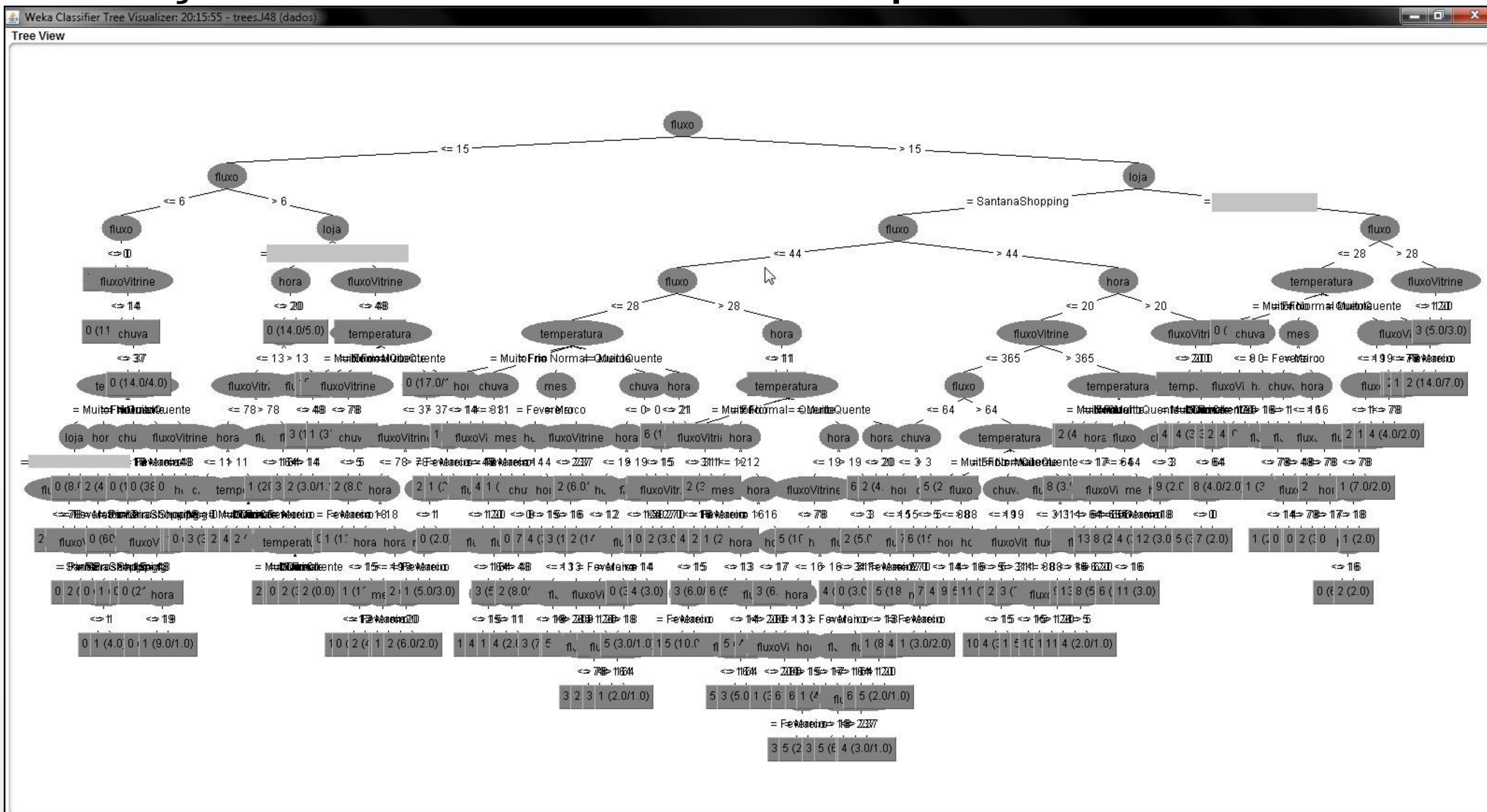
```
@relation dados

@attribute loja {ShoppingA, ShoppingB}
@attribute diaSemana {segunda, terca, quarta, quinta, sexta, sabado, domingo}
@attribute diaMes real
@attribute mes {Fevereiro, Marco}
@attribute hora real
@attribute fluxo real
@attribute fluxoVitrine real
@attribute temperatura real
@attribute umidade real
@attribute nebulosidade real
@attribute chuva real
@attribute atendimento {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28}

@data
ShoppingA,quarta,1,Fevereiro,10,15,193,23.4,79,10,1.0,0
ShoppingA,quarta,1,Fevereiro,11,36,67,23.4,79,10,1.0,4
ShoppingA,quarta,1,Fevereiro,12,38,93,22.0,82,10,1.0,4
ShoppingA,quarta,1,Fevereiro,13,33,269,22.0,82,10,1.0,3
ShoppingA,quarta,1,Fevereiro,14,66,229,22.0,82,10,1.0,1
ShoppingA,quarta,1,Fevereiro,15,72,441,22.0,82,10,1.0,2
ShoppingA,quarta,1,Fevereiro,16,54,326,22.0,82,10,1.0,3
ShoppingA,quarta,1,Fevereiro,17,63,23,22.0,82,10,1.0,5
ShoppingA,quarta,1,Fevereiro,18,52,241,26.5,61,10,1.0,3
ShoppingA,quarta,1,Fevereiro,19,51,224,26.5,61,10,1.0,5
ShoppingA,quarta,1,Fevereiro,20,40,388,26.5,61,10,1.0,10
ShoppingA,quarta,1,Fevereiro,21,157,126,26.5,61,10,1.0,7
ShoppingA,quarta,1,Fevereiro,22,69,165,26.5,61,10,1.0,0
```

Implementação

Geração da árvore de decisão pelo Weka:



Implementação

Classificação dos dados – Temperatura, Chuva e Fluxo:

Tipo de dado	Intervalo de valores	Resultado
Temperatura	18.0 a 21.0	Frio
Temperatura	21.2 a 24.0	Morno
Temperatura	24.2 a 26.8	Quente
Temperatura	27.0 a 33.6	MuitoQuente
Tipo de dado	**Intervalo de valores**	**Resultado**
Chuva	0.0	SemChuva
Chuva	0.1 a 1.0	PoucaChuva
Chuva	3.2 a 4.6	ChuvaLeve
Chuva	4.7 a 8.2	ChuvaModerada
Chuva	12.2 a 26.5	ChuvaForte
Chuva	31.0 a 46.6	Tempestade
Tipo de dado	**Intervalo de valores**	**Resultado**
Fluxo	0.0	SemFluxo
Fluxo	1 a 50	FluxoBaixo
Fluxo	51 a 100	FluxoModerado
Fluxo	101 a 200	FluxoAlto

Implementação

Classificação dos dados – Fluxo vitrine, Período e Atendimento:

Tipo de dado	Intervalo de valores	Resultado
Fluxo vitrine	0.0	VitrineNenhumMovimento
Fluxo vitrine	1 a 90	VitrineBaixo
Fluxo vitrine	91 a 300	VitrineMedio
Fluxo vitrine	300 a 743	VitrineAlto
Tipo de dado	**Intervalo de valores**	**Resultado**
Período	9 a 12	Manha
Período	12 a 18	Tarde
Período	19 a 22	Noite
Tipo de dado	**Intervalo de valores**	**Resultado**
Atendimento	0.0	SemAtendimento
Atendimento	1 a 3	1-3Atendimentos
Atendimento	4 a 6	4-6Atendimentos
Atendimento	7 a 9	7-9Atendimentos
Atendimento	10 a 12	10-12Atendimentos
Atendimento	13 a 15	13-15Atendimentos
Atendimento	19.0	19Atendimentos

Implementação

Classificação dos dados – Arquivo ARFF:

```
@relation dados

@attribute Loja {ShoppingA, ShoppingB}
@attribute DiaDaSemana {segunda-feira, terca-feira, quarta-feira, quinta-feira, sexta-feira, Sabado, domingo}
@attribute Mes {Fevereiro, Marco}
@attribute PeriodoDoDia {Manha, Tarde, Noite}
@attribute Fluxo {SemFluxo, FluxoBaixo, FluxoModerado, FluxoAlto}
@attribute FluxoDaVitrine {VitrineNenhumMovimento, VitrineBaixo, VitrineMedio, VitrineAlto}
@attribute Temperatura {Frio, Morno, Quente, MuitoQuente}
@attribute PrevisaoDoTempo {SemChuva, PoucaChuva, ChuvaLeve, ChuvaModerada, ChuvaForte, Tempestade}
@attribute Atendimentos {SemAtendimento, 1-3Atendimentos, 4-6Atendimentos, 7-9Atendimentos, 10-12Atendimentos

@data
ShoppingA,quarta-feira,Fevereiro,Manha,FluxoBaixo,VitrineMedio,Morno,PoucaChuva,SemAtendimento
ShoppingA,quarta-feira,Fevereiro,Manha,FluxoBaixo,VitrineBaixo,Morno,PoucaChuva,4-6Atendimentos
ShoppingA,quarta-feira,Fevereiro,Manha,FluxoBaixo,VitrineMedio,Morno,PoucaChuva,4-6Atendimentos
ShoppingA,quarta-feira,Fevereiro,Tarde,FluxoBaixo,VitrineMedio,Morno,PoucaChuva,1-3Atendimentos
```

Implementação

Desenvolvimento do webservice:

- Utilizado o estilo arquitetural Representational State Transfer (REST);
- Utilizado o JavaScript Object Notation (JSON) para transferência dos dados;
- Requisição Hypertext Transfer Protocol (HTTP);
- Carregamento do arquivo ARFF;
- Execução do algoritmo J48 da biblioteca da ferramenta Weka;

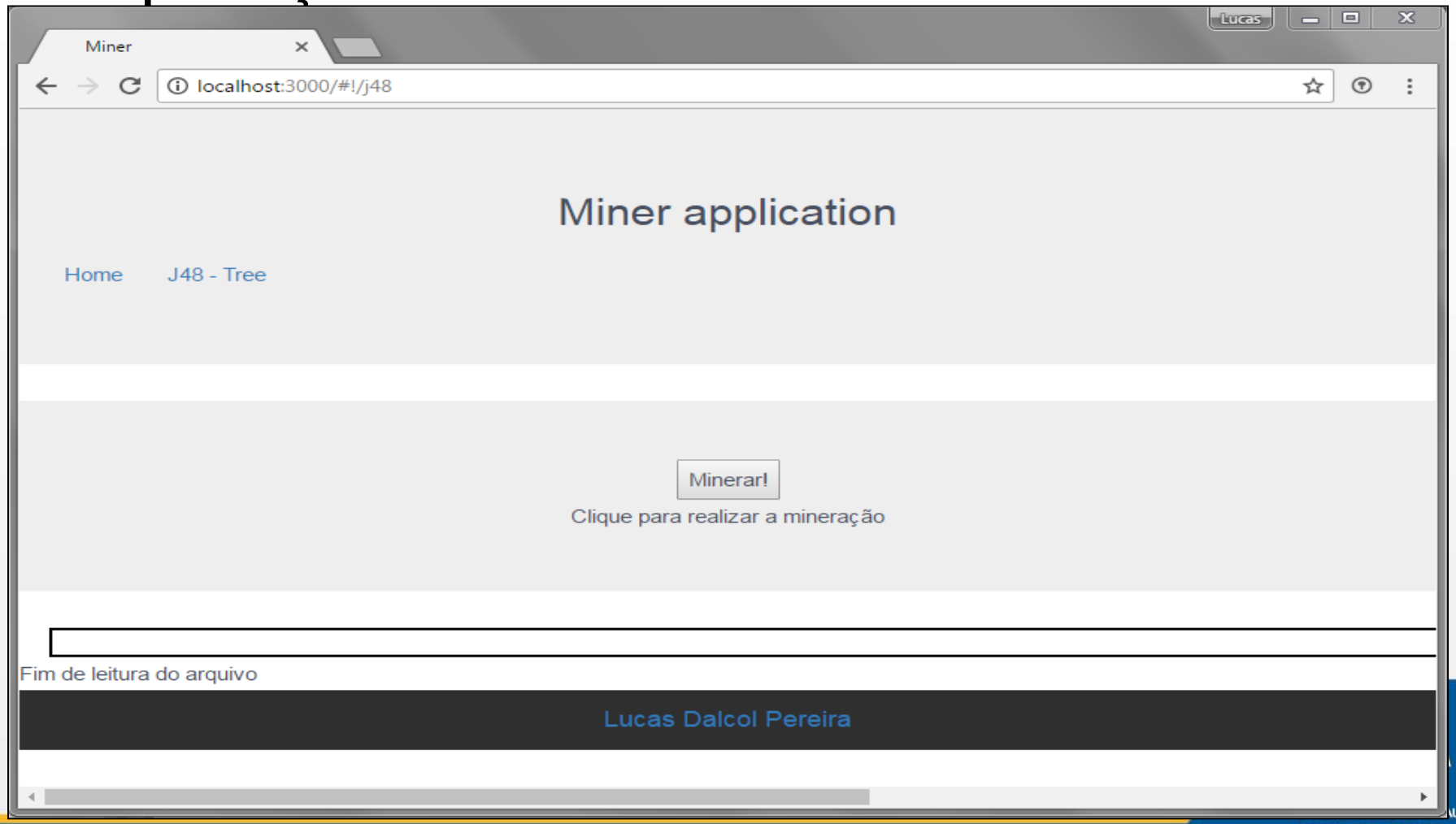
Implementação

Desenvolvimento do webservice:

- Obtenção da árvore gerada pelo algoritmo;
- Formatação dos dados para apresentação ao usuário;
- Cálculo do percentual de assertividade dos nós responsáveis pela quantidade de atendimento;
- Montagem dos dados da árvore e retornado para a requisição.

Operacionalidade da Implementação

- Aplicação – J48 - Tree:



Operacionalidade da Implementação

- Aplicação – Árvore de decisão:

Resultados

Análise da árvore de decisão:

Característica / árvore gerada	Ferramenta Weka	Software desenvolvido
Utilização do algoritmo J48	Sim	Sim
Utilização do arquivo ARFF gerado	Sim	Sim
Número de nós gerados	90	90
Nível mais profundo da árvore	7	7
Apresentação da árvore de decisão	Sim	Sim
Formato de apresentação da árvore	Horizontal	Vertical
Simplificação dos textos apresentados	Não	Sim

O software desenvolvido difere em duas características da ferramenta Weka e que permitem uma melhor compreensão e utilização do usuário.

Resultados

Comparação entre o software desenvolvido e seus correlatos:

Característica / trabalhos	Pereira (2017)	Juste (2013)	Gerosa (2011)	Librelotto (2013)
Gera árvore de decisão	Sim	Sim	Sim	Não
Utiliza arquivo ARFF para mineração	Sim	Sim	Não	Não
Utiliza banco de dados para mineração	Não	Não	Não	Sim
Realiza as etapas de KDD	Sim	Sim	Não	Sim
Utiliza o algoritmo J48	Sim	Sim	Não	Sim
Realiza a geração de gráficos	Não	Não	Sim	Não
Realiza comparação entre algoritmos	Não	Sim	Não	Sim
Utiliza dados reais para base de conhecimento	Sim	Não	Sim	Não
Aplica os resultados em situação real	Não	Não	Não	Não

Conclusões

- O objetivo de desenvolver um software para auxiliar neste processo permitindo um ganho estratégico para as empresas, foi atingido utilizando o algoritmo J48 disponibilizado pela ferramenta Weka e a geração da árvore de decisão;
- É possível identificar os fatores que influenciam no número de vendas e a quantidade esperada para cada combinação de atributos;

Conclusões

- O trabalho não foi submetido a empresas de vendas de varejo para validação dos dados obtidos;
- Para a obtençãode informações mais relevantes, seria necessário cruzar as informações extratificadas neste trabalho com os dados dos clientes que realizaram as compras. Desta forma será possível obter o perfil de cada cliente. Visando minimizar os gastos em armazenamento, transporte e compra;

Conclusões

- O processo de KDD é contínuo, os dados devem ser regularmente atualizados e aprimorados pelas etapas de pré-processamento e transformação, para que seja possível obter informações mais assertivas.

Sugestões

- Realizar a validação dos resultados obtidos, em uma empresa real;
- Permitir a seleção do arquivo que deve ser minerado;
- Permitir a edição do arquivo para mineração: desenvolver uma tela na aplicação para que o usuário possa importar um arquivo xls com os dados de mineração, gerar um arquivo ARFF e permitir que ao usuário e edição do arquivo;
- Execução de outros algoritmos para geração da árvore de decisão;

Referências

- DANTAS, Eric Rommel G. et al. **O uso da descoberta de conhecimento em base de dados para apoiar a tomada de decisões**. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA - SEGET, 5, 2008, João Pessoa - PB. Artigo. Resende - RJ: Aedb, 2008. p. 1 - 10. Disponível em: <http://www.aedb.br/seget/arquivos/artigos08/331_331_Artigo_SEGET_EJDR_Versao_Final_010808.pdf>. Acesso em: 10 mar. 2017.
- LIBRELOTTO, Solange Rubert; MOZZAQUATRO, Patricia Mariotto. Análise dos algoritmos de mineração j48 e apriori aplicados na detecção de indicadores da qualidade de vida e saúde. **Revint: Revista Interdisciplinar de Ensino, Pesquisa e Extensão**, Cruz Alta - RS, v. 1, n. 1, p.1-37, jun. 2013. Disponível em: <http://www.revistaeletronica.unicruz.edu.br/index.php/eletronica/article/viewFile/26-37/pdf>. Acesso em: 20 jun. 2017.
- MACEDO, Dayana Carla de; MATOS, Simone Nasser. Extração de conhecimento através da mineração de dados. **Revista de Engenharia e Tecnologia**, Curitiba - PR, v. 2, n. 2, p.23-29, ago. 2010. Disponível em: <http://www.revistaret.com.br/ojs-2.2.3/index.php/ret/article/viewFile/38/73>. Acesso em: 20 jun. 2017.
- REZENDE, Solange Oliveira. **Sistemas Inteligentes**: Fundamentos e Aplicações. Barueri - SP: Manole Ltda, 2003.
- WU, Xindong et al. **Top 10 algorithms in data mining**. Knowledge And Information Systems, [s.l.], v. 14, n. 1, p.1-37, 4 dez. 2007. Springer Nature. http://dx.doi.org/10.1007/s10115-007-0114-2. Disponível em: <http://www.cs.umd.edu/~samir/498/10Algorithms-08.pdf>. Acesso em: 20 jun. 2017.