

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Programa de Pós-Graduação em Ciência da Computação

SPDW-Miner: UM MÉTODO PARA
A EXECUÇÃO DE PROCESSOS
DE DESCOBERTA DE
CONHECIMENTO EM BASES
DE DADOS DE
PROJETOS DE *SOFTWARE*

Fernanda Vieira Figueira

Dissertação de Mestrado

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz

Porto Alegre
2008

Fernanda Vieira Figueira

***SPDW-Miner: UM MÉTODO PARA
A EXECUÇÃO DE PROCESSOS
DE DESCOBERTA DE
CONHECIMENTO EM BASES
DE DADOS DE
PROJETOS DE SOFTWARE***

**Dissertação apresentada como
requisito para obtenção do grau de
Mestre pelo Programa de Pós-
graduação da Faculdade de
Informática da Pontifícia
Universidade Católica do Rio Grande
do Sul.**

Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz

Porto Alegre
2008

Dados Internacionais de Catalogação na Publicação (CIP)

F475S Figueira, Fernanda Vieira
SPDW - Miner : um método para execução de processos de descoberta de conhecimento em bases de dados de projetos de software / Fernanda Vieira Figueira. – Porto Alegre, 2008.
98 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS
Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz.

1. Informática. 2. Mineração de Dados (Informática).
3. Data Warehouse. I. Ruiz, Duncan Dubugras Alcoba Ruiz.
II. Título.

CDD 005.74

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**SPDW-Miner: Um Método para a Execução de Processos de Descoberta de Conhecimento em Bases de Dados de Projetos de Software**", apresentada por Fernanda Vieira Figueira, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 31/03/08 pela Comissão Examinadora:

Prof. Dr. Duncan Dubugras Alcoba Ruiz –
Orientador

PPGCC/PUCRS

Profa. Dra. Vera Lúcia Strube de Lima –

PPGCC/PUCRS

Profa. Dra. Viviane Moreira Orengo –

UFRGS

Homologada em 04/08/09, conforme Ata No. 013/09 pela Comissão Coordenadora.

Prof. Dr. Fernando Gehm Moraes
Coordenador.



PUCRS

Campus Central

Av. Ipiranga, 6681 – P32 – sala 507 – CEP: 90619-900

Fone: (51) 3320-3611 – Fax (51) 3320-3621

E-mail: ppgcc@inf.pucrs.br

www.pucrs.br/facin/pos

AGRADECIMENTOS

Agradeço a Deus por ter me dado à oportunidade de chegar até aqui!!

A minha família pelo carinho e incentivo que me deram ao longo dos dois anos de mestrado. Não tenho palavras para agradecer tudo!!

Aos meus orientadores, Duncan e Karin, pelo apoio incondicional e pela paciência para concretizar este sonho! Muito obrigada, mesmo!

Aos meus grandes amigos, Hugo e Márcio, por terem confiado em mim e me ajudado a ingressar na PUC.

À Aninha e a Tita, que não foram apenas colegas de mestrado, mas sim verdadeiras amigas!!

Ao amigo Ezequiel pelas palavras de conforto, carinho e ajuda prestada para finalizar este trabalho.

A todos os amigos e colegas, que não citei aqui, mas que me apoiaram de alguma forma nesta conquista.

Agradeço, também, ao convênio PUCRS (PPGCC-PUCRS) e a HP EAS Brasil por terem me concedido a bolsa que custeou os meus estudos.

*“Se chorei
Ou se sorri
O importante é que emoções eu vivi...”*
Roberto Carlos e Erasmo Carlos

RESUMO

As organizações de *software* buscam, cada vez mais, aprimorar seu Processo de Desenvolvimento de *Software* (PDS), com o intuito de garantir a qualidade dos seus processos e produtos. Para tanto, elas adotam modelos de maturidade de *software*. Esses modelos estabelecem que a mensuração da qualidade seja realizada através de um programa de métricas (PM). As métricas definidas devem ser coletadas e armazenadas, permitindo manter um histórico organizacional da qualidade.

Contudo, apenas mensurar não é o bastante. As informações armazenadas devem ser úteis para apoiar na manutenção da qualidade do PDS. Para tanto, os níveis mais altos dos modelos de maturidade sugerem que técnicas estatísticas e analíticas sejam utilizadas, com a finalidade de estabelecer o entendimento quantitativo sobre as métricas. As técnicas de mineração de dados entram neste contexto como uma abordagem capaz de aumentar a capacidade analítica e preditiva sobre as estimativas e o desempenho quantitativo do PDS.

Este trabalho propõe um método para a execução do processo de *KDD* (Knowledge Discovery in Database), denominado de *SPDW-Miner*, voltado para a predição de métricas de *software*. Para tanto, propõe um processo de *KDD* que incorpora o ambiente de data warehousing, denominado *SPDW+*. O método é composto por uma série de etapas que guiam os usuários para o desenvolvimento de todo o processo de *KDD*. Em especial, em vez de considerar o *DW* (data warehouse) como um passo intermediário deste processo, o toma como ponto de referência para a sua execução. São especificadas todas as etapas que compõem o processo de *KDD*, desde o estabelecimento do objetivo de mineração; a extração e preparação dos dados; a mineração até a otimização dos resultados. A contribuição está em estabelecer um processo de *KDD* em um nível de detalhamento bastante confortável, permitindo que os usuários organizacionais possam adotá-lo como um manual de referência para a descoberta de conhecimento.

Palavras Chave: Processo de *KDD*, Técnica de Classificação, Métricas de *Software e Data warehouse*.

ABSTRACT

Software organizations aim at improving their *Software Development Process* (SDP) targeting the quality assessment of their processes and products. They adopt *software* maturity models to achieve this. Maturity models define quality measuring should be done through a metrics program. The defined metrics must be collected and stored properly, maintaining the history of the organizational quality data.

However, measuring alone is not enough. Stored data must be useful to support SDP quality maintenance. To do that, maturity models suggest the use of statistical and analytical techniques. The goal is to make feasible the quantitative understanding of the metrics. Data mining techniques are useful in this scenario as an approach able to improve analytical and predictive capabilities on estimations and performance of SDP.

This work introduces a method of performing *KDD* process, named *SPDW-Miner*, oriented to *software* metrics prediction. It is proposed a *KDD* process that incorporates the *SPDW+* data-warehousing environment. Such method is composed by a set of steps that guide users to apply the whole *KDD* process. In special, instead of considering *DW* as an intermediate step, *SPDW-Miner* adopts it as a reference to rule its execution. It is specified all *KDD* process steps: defining the mining goal; extracting a preparing data; data mining and results optimization. The contribution of this work is the establishing of a *KDD* process, in a proper, user-comfortable detail level. It enables organizational users can to adopt it as a reference guide to knowledge discovery.

Keywords: *Knowledge Discovery in Databases, Classification, Software Metrics and Data warehouse.*

LISTA DE FIGURAS

<i>Figura 1: Estrutura da Representação em Estágios do CMMI [SEI06].....</i>	<i>19</i>
<i>Figura 2: Processo de descoberta de conhecimento (KDD) [HAN01].</i>	<i>24</i>
<i>Figura 3: Arquitetura do SPDW+ [SIL07].....</i>	<i>34</i>
<i>Figura 4: Modelo Analítico do SPDW+ [SIL07].....</i>	<i>35</i>
<i>Figura 5: Processo de KDD voltado para predição de métricas de software.....</i>	<i>40</i>
<i>Figura 6: Etapas do SPDW-Miner.....</i>	<i>45</i>
<i>Figura 7: Resultado de uma consulta na base de dados do ClearQuest.</i>	<i>57</i>
<i>Figura 8: Categorias do atributo classe.</i>	<i>59</i>
<i>Figura 9: Trecho dos modelos preditivos obtidos nos experimentos 1 e 2.</i>	<i>60</i>
<i>Figura 10: Matriz de Confusão do experimento 3.</i>	<i>62</i>
<i>Figura 11: Categorias do Atributo Número de Colaboradores.....</i>	<i>63</i>
<i>Figura 12: Trecho dos modelos preditivos obtidos nos experimentos 3 e 4.</i>	<i>66</i>
<i>Figura 13: Trecho dos modelos preditivos obtidos nos experimentos 5 e 6.</i>	<i>66</i>
<i>Figura 14: Trecho dos modelos preditivos obtidos nos experimentos 7 e 8.</i>	<i>67</i>
<i>Figura 15: Trecho do modelo preditivo obtido no experimento 9.</i>	<i>70</i>
<i>Figura 16: Trecho do modelo preditivo obtido no experimento 10.</i>	<i>71</i>
<i>Figura 17: Trecho dos modelos preditivos obtidos nos experimentos 11 e 12.</i>	<i>71</i>
<i>Figura 18: Trecho dos modelos preditivos obtidos nos experimentos 13 e 14.</i>	<i>71</i>
<i>Figura 19: Trecho dos modelos preditivos obtidos nos experimentos 15 e 16.</i>	<i>75</i>
<i>Figura 20: Trecho dos modelos preditivos obtidos nos experimentos 17 e 18.</i>	<i>75</i>
<i>Figura 21: Trecho dos modelos preditivos obtidos nos experimentos 19 e 20.</i>	<i>76</i>
<i>Figura 22: Trecho dos modelos preditivos obtidos nos experimentos 21 e 22.</i>	<i>79</i>
<i>Figura 23: Trecho dos modelos preditivos obtidos nos experimentos 23 e 24.</i>	<i>79</i>
<i>Figura 24: Trechos dos modelos preditivos obtidos nos experimentos 25 e 26.</i>	<i>81</i>
<i>Figura 25: Trecho do modelo preditivo obtido no experimento 27 após a otimização.</i>	<i>84</i>
<i>Figura 26: Modelo preditivo obtido no experimento 28 após a otimização.</i>	<i>84</i>
<i>Figura 27: Trecho do modelo preditivo obtido no experimento 29 após a otimização.</i>	<i>84</i>

LISTA DE TABELAS

<i>Tabela 1: Áreas de Processo por nível no Modelo CMMI.....</i>	<i>22</i>
<i>Tabela 2: Comparativo entre as diferentes propostas do processo de KDD.....</i>	<i>24</i>
<i>Tabela 3: Matriz de Confusão.....</i>	<i>29</i>
<i>Tabela 4: Programa de Métricas do SPDW+ [SIL07].....</i>	<i>36</i>
<i>Tabela 5: Comparações entre os trabalhos relacionados.</i>	<i>91</i>

LISTA DE SIGLAS

%TC	% Trabalho Completado
AP	Área de Processo
ARFF	<i>Attribute-relation file format</i>
BI	<i>Business Intelligence</i>
BO	Base Organizacional
CART	<i>Classification And Regression Trees</i>
CBO	Custo <i>Baseline</i> Original
CBR	Custo <i>Baseline</i> Revisado
CMM	<i>Capability Maturity Model</i>
CMMI	<i>Capability Maturity Model Integration</i>
CR	Custo Real
CRAR	Custo Real da Atividade de Revisão
CRAQ	Custo Real da Atividade de Qualidade
CRFT	Custo Real da Fase de Teste
DDE	Densidade de Defeitos Externos
DDI	Densidade de Defeitos Internos
DFBO	Data Final <i>Baseline</i> Original
DFBR	Data Final <i>Baseline</i> Revisado
DFR	Data Final Real do Projeto
DIBO	Data Inicial <i>Baseline</i> Original
DIBR	Data Inicial <i>Baseline</i> Revisado
DS	Data <i>Status</i>
DTS	<i>Data Transformation Services</i>
DW	<i>Data warehouse</i>
EBO	Esforço <i>Baseline</i> Original
EBR	Esforço <i>Baseline</i> Revisado
ER	Esforço Real
ERD	Eficiência de Remoção de Defeitos
ERV	Eficiência de Revisão
ETC	Extração, Transformação e Carga
EVA	<i>Earned Value Analysis</i>
FN	Falsos Negativos
FP	Falsos Psitivos

IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISC	Índice de Satisfação do Cliente
KDD	<i>Knowledge Discovery in Databases</i>
NDI	Número de Defeitos Internos
NDE	Número de Defeitos Externos
NMA	Número de Modificações Aprovadas
NMR	Número de Modificações Requeridas
OLAP	<i>On-Line Analytical Process</i>
OSSP	<i>Organization's Set of Standard Process</i>
PDS	Processo de Desenvolvimento de <i>Software</i>
PM	Programa de Métricas
PR	Produtividade
SDP	<i>Software Development Process</i>
SPDW+	<i>SDP Performance Data warehousing Plus</i>
SQL	<i>Structured Query Language</i>
TBO	Tamanho <i>Baseline</i> Original
TBR	Tamanho <i>Baseline</i> Revisado
TI	Tecnologia de Informação
TR	Tamanho Real
VA	Valor Agregado
VBO	Varição do <i>Baseline</i> Original
VBR	Varição do <i>Baseline</i> Revisado
VC	Varição de Custo
VCA	Varição de Custo Agregada
VE	Valor Estimado
VEBO	Varição de Esforço do <i>Baseline</i> Original
VEBR	Varição de Esforço do <i>Baseline</i> Revisado
VP	Varição de Prazo
VPA	Varição de Prazo Agregada
VR	Volatilidade de Requisitos
VN	Verdadeiros Negativos
VPo	Verdadeiros Positivos
VTBO	Varição de Tamanho do <i>Baseline</i> Original
VTBR	Varição de Tamanho do <i>Baseline</i> Revisado

SUMÁRIO

1	INTRODUÇÃO	15
2	REFERENCIAL TEÓRICO	17
2.1	MÉTRICAS DE SOFTWARE	17
2.1.1	<i>Programa de Métricas (PM)</i>	18
2.2	MODELO DE MATURIDADE DE SOFTWARE	18
2.2.1	<i>Áreas de Processo do CMMI</i>	20
2.3	DATA WAREHOUSE	22
2.4	PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS.....	23
2.4.1	<i>Pré-processamento de Dados</i>	25
2.4.2	<i>Mineração de Dados</i>	26
2.4.2.1	Classificação	27
2.4.2.2	Classificação por árvore de decisão.....	27
2.4.2.3	Critério de Avaliação de Modelos de Classificação.....	28
2.4.3	<i>Pós-processamento de Dados</i>	29
3	DESCRIÇÃO DO CENÁRIO	31
3.1	PROBLEMÁTICA	31
3.1.1	<i>Análise e Monitoração</i>	31
3.1.2	<i>Previsão</i>	32
3.2	SPDW+	33
3.2.1	<i>Repositório de Dados</i>	34
3.2.2	<i>Métricas potencialmente úteis para predição</i>	36
3.2.3	<i>Considerações sobre o SPDW+</i>	37
3.3	CENÁRIO REAL ESTUDADO.....	37
3.3.1	<i>Repositório de Métricas</i>	37
3.3.2	<i>Processo de Carga</i>	38
3.3.3	<i>Interface de BI</i>	39
3.3.4	<i>Considerações sobre o cenário real estudado</i>	39
3.4	CARACTERIZAÇÃO DA CONTRIBUIÇÃO.....	39
3.5	CONSIDERAÇÕES SOBRE A DESCRIÇÃO DO CENÁRIO	42
4	SPDW-MINER – MÉTODO DE DESCOBERTA DE CONHECIMENTO	43
4.1	PERFIL DO USUÁRIO	43
4.2	ETAPAS DO MÉTODO SPDW-MINER	44
4.3	CONSIDERAÇÃO SOBRE O MÉTODO SPDW-MINER.....	54
5	ESTUDO DE CASO	55
5.1	OBJETIVO DOS EXPERIMENTOS	55
5.2	EXPERIMENTOS.....	55
5.2.1	<i>Experimentos com Categorização em 10 faixas de valores</i>	58
5.2.1.1	Discussão dos resultados com Categorização em 10 faixas de valores.....	60
5.2.2	<i>Experimentos com Categorização em 13 faixas de valores</i>	60
5.2.2.1	Discussão dos resultados com Categorização em 13 faixas de valores.....	65
5.2.3	<i>Experimentos com Categorização em 9 faixas de valores</i>	67
5.2.3.1	Discussão dos Resultados com Categorização em 9 faixas de valores	70

5.2.4	<i>Experimentos com Categorização em 4 faixas de valores (']inf-2]', ']</i>	
	<i>4]', ']4-8]', ' > 8Horas')</i>	72
5.2.4.1	Discussão dos Resultados com Categorização em 4 faixas de valores (']inf-	
	2]', ']2-4]', ']4-8]', ' >	
	8Horas').....	75
5.2.5	<i>Experimentos com Categorização em 4 faixas de valores (']-inf-1]', ']1-2]',</i>	
	<i>]2-3]', ' > 3 Horas')</i>	76
5.2.5.1	Discussão dos Resultados com a Categorização em 4 faixas de valores (']-	
	inf-1]', ']1-2]', ']2-3]', ' > 3 Horas').....	78
5.2.6	<i>Experimentos com Categorização em 2 faixas de valores</i>	79
5.2.6.1	Discussão dos Resultados com Categorização em 2 faixas de valores	81
5.2.7	<i>Experimentos com Categorização em 4 faixas de valores (']-inf-1.5]', ']1.5 –</i>	
	<i>4]', ']4-6]', ' > 6 Horas')</i>	81
5.2.7.1	Discussão dos Resultados com Categorização em 4 faixas de valores (']-inf-	
	1.5]', ']1.5 – 4]', ']4-6]', ' > 6 Horas').....	83
5.3	CONSIDERAÇÕES SOBRE O ESTUDO DE CASO.....	85
6	TRABALHOS RELACIONADOS	87
6.1	NAYAK E QIU [NAY05]	87
6.2	KHOSHGOFTAAR ET AL. [KHO01].....	88
6.3	NAGAPPAN ET AL. [NAG06]	89
6.4	WINCK [WIN07]	90
6.5	CONSIDERAÇÕES SOBRE OS TRABALHOS RELACIONADOS	90
7	CONSIDERAÇÕES FINAIS	93
7.1	TRABALHOS FUTUROS	93
	REFERÊNCIAS BIBLIOGRÁFICAS	95
	APÊNDICE A – DIAGRAMA DO MODELO ANALÍTICO	98

1 INTRODUÇÃO

As organizações de software buscam, cada vez mais, aprimorar seu Processo de Desenvolvimento de Software, com o intuito de garantir a qualidade dos seus produtos. Segundo [KHO01], a qualidade do produto de software está diretamente relacionada à qualidade do seu processo de desenvolvimento. Desta forma, na tentativa de assegurar a qualidade do PDS, muitas empresas estão adotando modelos de maturidade de software, como o CMMI (Capability Maturity Model Integration) [SEI06] e o MPS.Br (Melhoria de Processos do Software Brasileiro) [SOF07]. Esses modelos definem os elementos necessários para tornar o PDS definido, eficiente e controlado, através de etapas evolutivas do processo.

O *CMMI* estabelece que as organizações devem definir um conjunto de processos padrão, chamado de *OSSP (Organization's Set of Standard Process)*, o qual pode ser especializado para refletir as particularidades dos diferentes projetos de desenvolvimento de *software*. Após os processos serem definidos, eles devem ser mensurados para permitir o seu controle e assegurar a sua qualidade. A mensuração é realizada através do estabelecimento de um Programa de Métricas (PM). Para que uma organização de *software* seja certificada com o *CMMI* nível 3, esta deve apresentar um programa de métricas definido e um repositório para armazená-las. No entanto, compreender o complexo relacionamento entre as métricas e os demais atributos do PDS, para controlar a qualidade, não é uma tarefa trivial [DIC04]. O *CMMI* nível 4 prevê a utilização de técnicas estatísticas e analíticas sobre as métricas para estabelecer o entendimento quantitativo dos processos. Esse entendimento é fornecido através do estabelecimento de modelos de desempenho de processo, os quais são usados para representar comportamentos passados e atuais, assim como para prever futuros resultados dos processos [SEI06]. As técnicas de mineração de dados entram neste contexto como uma abordagem capaz de aumentar a capacidade analítica e preditiva sobre as estimativas e o desempenho quantitativo do PDS.

Este trabalho aborda, justamente, o uso de técnica de classificação para predição de métricas de *software*. Para tanto, propõe um processo de *KDD* que incorpora o ambiente de *data warehousing*, denominado *SPDW+*, para previsão de métricas de *software*. Em especial, um método de execução do processo de *KDD* que, em vez de considerar o *DW* como um passo intermediário deste processo, o toma como ponto de referência para a sua execução.

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta o referencial teórico, o capítulo 3 a descrição do cenário, o capítulo 4 relata o método de execução do

processo de *KDD*, o capítulo 5 apresenta o estudo de caso onde é aplicado o método, o capítulo 6 descreve os trabalhos relacionados e o capítulo 7 discorre sobre as considerações finais e os trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos e assuntos relacionados ao tema de pesquisa. Para tanto, o mesmo aborda uma breve explanação sobre: (i) métricas de *software*, (ii) modelo de maturidade de *software*, (iii) *data warehouse*, e (iv) processo de descoberta de conhecimento em base de dados.

2.1 Métricas de Software

Segundo o padrão IEEE 1061 [IEE98], a qualidade de *software* é o grau no qual é apresentada uma combinação desejada de atributos. Este padrão estabelece requisitos de qualidade, através da identificação, implementação, análise e validação de métricas de qualidade de produtos e processos de *software*. A fim de medir os atributos da qualidade do *software*, um conjunto apropriado de métricas deve ser identificado e definido.

Uma métrica [IEE98] é uma função mensurável, cujas entradas são dados de *software* e seu resultado corresponde a um único valor numérico, que pode ser interpretado como o grau de qualidade do *software*, sendo esse afetado por determinado atributo. O objetivo das métricas é avaliar o produto de *software* durante todo o seu ciclo de vida, em relação às exigências de qualidade que foram definidas. As métricas permitem controlar a qualidade do processo e dos produtos desenvolvidos [GOP02]. O uso delas em uma organização ou projeto permite uma melhor visibilidade da qualidade.

A definição de um conjunto de métricas deve ser realizada de acordo com as reais necessidades da organização. Os modelos de maturidade prevêm que métricas de produto e processo sejam definidas. O padrão *IEEE* 1061 [IEE98], define métricas de *software* em duas categorias: de produto e de processo. As métricas de produto são usadas para medir as características de uma documentação ou código (*e.g.* tamanho do produto e complexidade do fluxo de dados). As de processo são utilizadas para medir características do processo e do ambiente de desenvolvimento (*e.g.* eficiência de remoção de defeitos e experiência do programador). [KAN03] utiliza a mesma classificação para as métricas que o padrão *IEEE* 1061 [IEE98], porém prevê mais uma categoria de métrica, as de projeto. As métricas de projeto definem características do projeto e de sua execução (*e.g.* número de recursos e produtividade).

O padrão IEEE 1061 também prevê a classificação de métricas como diretas e indiretas. As diretas são aquelas que não dependem de nenhum outro atributo. E as indiretas são calculadas em função de outros atributos.

A utilização de métricas de software reduz a subjetividade da avaliação da qualidade, pois fornece uma base quantitativa para a tomada de decisão sobre a qualidade do software. Entretanto, o uso de métricas não elimina a necessidade do julgamento humano na avaliação do software [IEE98].

2.1.1 Programa de Métricas (PM)

Um PM é uma forma de documentar e organizar as métricas, permitindo que o uso delas seja fomentado e institucionalizado em uma organização [SEI06]. A norma IEEE 1061 e os modelos de maturidade como *CMM* e *CMMI* sugerem a definição e utilização de um PM que seja significativo para os projetos e para a organização. Contudo, não definem quais métricas devem ser utilizadas. Estas devem ser estabelecidas através das reais necessidades da organização, buscando atingir os diferentes níveis de gerenciamento organizacional. Para facilitar a análise e focar em todos os pontos de necessidades de informação elas podem ser, convenientemente, agrupadas em áreas de qualidade. Um PM para ser abrangente e eficiente deve cobrir todas as áreas de qualidade de uma empresa (*e.g.* defeitos, requisitos, cronograma e etc).

Em um PM além da definição de cada métrica, deve constar [IEE98]: (i) o seu nome; (ii) o custo, os benefícios e os impactos associados a sua utilização; (iii) as faixas de valores esperados; (iv) as ferramentas de armazenamento utilizadas; (v) a sua aplicabilidade; (vi) os valores de entrada necessários para o seu cálculo; e (vii) um exemplo de sua aplicação.

2.2 Modelo de Maturidade de *Software*

Segundo [SOM04], processo de desenvolvimento de *software* é um conjunto de atividades e resultados associados que levam à produção de um produto de *software*. Para [PRE04], este processo pode ser definido como uma estrutura para as tarefas que são necessárias à construção de um *software* de alta qualidade. A busca por qualidade no PDS tem impulsionado a adoção de modelos de maturidade de *software*, como *CMMI* e o MPS.Br. Nestes modelos a evolução do PDS é adquirida através de etapas evolutivas do processo, auxiliando as organizações de *software* no processo de seleção de estratégias de melhoria,

determinando a maturidade atual dos respectivos PDS, bem como, identificando questões críticas para seu aperfeiçoamento.

O *CMMI* é um modelo de maturidade que fornece orientações para melhorar os processos de uma organização e a habilidade de gerenciar o desenvolvimento, aquisição, manutenção de produtos e serviços [SEI06]. Existem dois tipos de representação do *CMMI*: contínua e em estágios, sendo que ambas possuem o mesmo conteúdo, diferindo apenas na estruturação.

A representação em estágios é a mesma utilizada no *CMM* (*Capability Maturity Model*) antecessor do *CMMI*, e a sua estrutura é mostrada na Figura 1. Nela são definidos cinco níveis de Maturidade (*Maturity Levels*), os quais são compostos de áreas de processo (*Process Areas*), objetivos genéricos (*Generic Goals*) e específicos (*Specific Goals*), práticas genéricas (*Generic Practices*) e específicas (*Specific Practices*). A representação em estágio é a abordada neste trabalho por ser a mais usual, esta é mostrada na Figura 1.

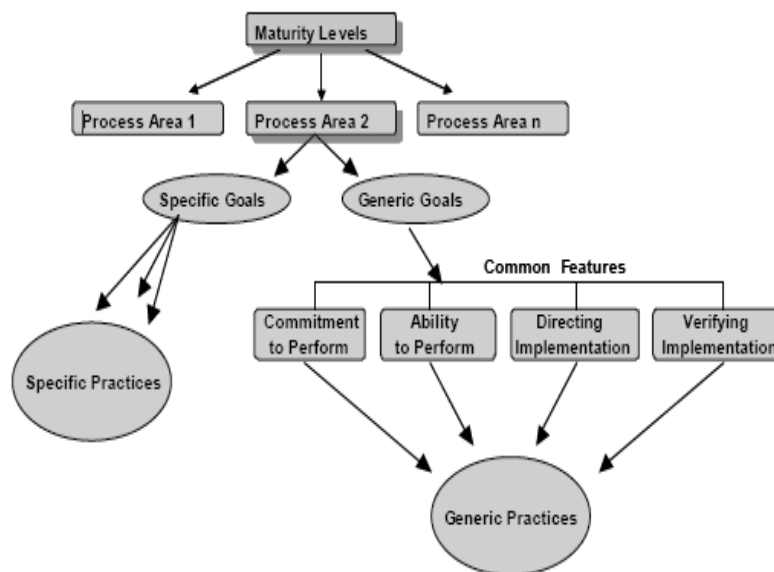


Figura 1: Estrutura da Representação em Estágios do CMMI [SEI06].

Os cinco níveis de maturidade da representação em estágio do *CMMI* são: (1) Inicial, (2) Gerenciado, (3) Definido, (4) Gerenciado Quantitativamente, e (5) Otimizado.

O nível 1 é caracterizado por não apresentar nenhum procedimento eficiente de gerência de projeto. O processo de desenvolvimento é *ad hoc* e caótico [SEI06].

No nível 2 os processos são planejados, executados, medidos e controlados. Assim,

existe a necessidade de se estabelecer um PM, que contemple as diferentes áreas de qualidade da organização.

No nível 3 os processos são bem definidos e entendidos, um conjunto de processos padrões da organização é estabelecido, e assim, cada projeto pode adaptá-los de acordo com a sua realidade. Além disso, existe a necessidade da implementação de um repositório de dados para armazenar as métricas definidas no nível 2. Esta base de dados permite uma visão unificada das informações de todos os projetos da organização, possibilitando suporte à decisão organizacional. Além de compartilhar boas práticas e experiências de desenvolvimento de *software*. Em muitos casos, esses repositórios de métricas baseiam-se em *data warehouse (DW)* ([SUB99], [PAL03], [BEC06] e [SIL07]), estruturados segundo um modelo analítico multidimensional e inseridos em um ambiente de *data warehousing*. Nestes ambientes além do repositório deve existir um processo de extração, transformação e carga (ETC) para extrair, preparar e carregar as métricas provenientes de várias fontes no repositório central, sendo esta etapa crucial para garantir a qualidade das métricas coletadas [KIM98]. Tanto as características desse modelo, como os conceitos que envolvem a definição de um *DW* encontram-se detalhadas na seção 2.3.

O nível 4 é o gerenciado quantitativamente. Nele os subprocessos são selecionados para serem gerenciados quantitativamente, através de técnicas estatísticas e analíticas. Neste nível os objetivos quantitativos para a qualidade são estabelecidos e utilizados como um critério de gerenciamento. A qualidade e o desempenho dos processos são entendidos em termos quantitativos, e controlados durante todo o ciclo de vida do projeto. Neste nível existe a necessidade dos processos serem previsíveis e, para tanto, técnicas voltadas para este fim devem ser usadas.

No nível 5 os processos são continuamente otimizados, através do entendimento quantitativo. Esse nível foca na contínua manutenção do desempenho dos processos através de inovações tecnológicas. Os objetivos de melhoramento quantitativos da organização são estabelecidos e continuamente revisados para refletir os diferentes objetivos de negócios.

2.2.1 Áreas de Processo do CMMI

Uma área de processo (AP) é um conjunto de práticas relacionadas que, desempenhadas coletivamente, satisfazem um conjunto de objetivos considerados importantes para alcançar um melhoramento significativo em uma determinada área [SEI06]. Para uma organização alcançar o nível 3 de maturidade, por exemplo, esta deve alcançar todos os

objetivos genéricos e específicos das áreas de processo do nível 2 e 3. A Tabela 1, apresenta todas as áreas de processo distribuídas por níveis.

Entre as AP do modelo *CMMI* são destacadas, neste trabalho, a de Mensuração e Análise (nível 2), Gerenciamento Quantitativo de Projeto e Desempenho de Processo Organizacional (nível 4). Em síntese, essas AP visam definir a mensuração dos processos para, assim, estabelecer o gerenciamento quantitativo.

A AP de Mensuração e Análise é uma das mais importantes do modelo *CMMI*. Ela tem a finalidade de desenvolver e sustentar a capacidade de mensuração, a qual é usada para dar suporte ao gerenciamento de informações. Ela provê práticas específicas que conduzem os projetos e a organização no estabelecimento de um PM e dos resultados que estas podem gerar para apoiar à tomada de decisão e ações corretivas do processo em tempo hábil. Para permitir a utilização das métricas como uma ferramenta de apoio à gestão dos processos, o *CMMI* aponta a necessidade da implementação de um repositório. Este último visa armazenar os processos organizacionais e suas métricas coletadas, de forma organizada e concisa, permitindo que os gestores possam acessá-lo para efeitos de análise. Através das métricas estabelecidas no nível 2 e do repositório no nível 3, é possível estimar e planejar atividades, prazos, custos, e ainda fornecer uma visão unificada e quantitativa sobre a qualidade dos projetos da organização.

As outras duas AP que se destacam são Gerenciamento Quantitativo de Projeto e Desempenho de Processo Organizacional, ambas do nível 4. Essas duas visam implantar e manter o entendimento quantitativo dos processos organizacionais, provendo suporte à qualidade, aos objetivos de desempenho e às estimativas e, ainda, a disponibilização de modelos para gerenciar quantitativamente os projetos organizacionais. O entendimento quantitativo é estabelecido através de modelos de desempenho, os quais são usados para representar desempenhos passados e atuais, e também para prever futuros resultados dos processos [SEI06]. As organizações usam essas informações quantitativas e técnicas analíticas para caracterizar produtos e processos. Essa caracterização é útil para [SEI06]:

- Determinar se os processos estão se comportando consistentemente ou se têm tendências estáveis, isto é, podem ser preditos;
- Identificar os processos onde o desempenho está dentro dos limites estabelecidos;
- Estabelecer critérios para identificar se um processo ou elemento dele está quantitativamente controlado, e determinar as medidas e técnicas analíticas pertinentes a tal gerência;

- Identificar os processos que se mostram anômalos;
- Identificar todos os aspectos dos processos que podem ser melhorados no conjunto de processos padrão da organização;
- Identificar os processos que têm o melhor desempenho.

Tabela 1: Áreas de Processo por nível no Modelo CMMI.

Nível de Maturidade	Áreas de Processos
5- Otimizado	Inovação e Desenvolvimento Organizacional; Análise e Resolução de Causas;
4- Gerenciado Quantitativamente	Desempenho do Processo Organizacional; Gerência Quantitativa de Projeto;
3- Definido	Desenvolvimento de Requisitos; Solução Técnica; Integração de Produto; Verificação; Validação; Gerência de Riscos; Gerência Integrada de Fornecedores; Gerência Integrada de Projeto; Definição do Processo da Organização; Foco no Processo da Organização; Treinamento Organizacional;
2- Gerenciado	Controle e Monitoração de Projetos; Garantia de Qualidade do Processo e Produto; Gerência de Configuração; Gerência de Acordo com Fornecedores; Gerência de Requisitos; Planejamento do Projeto; Mensuração e Análise;
1- Inicial	Não se aplica.

Desta forma, uma possível técnica analítica para permitir o estabelecimento dos modelos de desempenho dos processos é a técnica de mineração. As técnicas de mineração, como a classificação e a regressão, podem ser usadas para gerar os modelos de desempenho de processo, permitindo dar maior previsibilidade aos mesmos, através do estabelecimento de modelos preditivos. Esses modelos são úteis para justificar determinados comportamentos observados nos processos de *software*, realizar estimativas de prazo e custo. Ainda, pode-se prever possíveis falhas e mostrar, antecipadamente, desalinhos nos objetivos do projeto.

2.3 Data warehouse

Um *data warehouse* (DW) é um conjunto de dados baseado em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais [INM05]. O fato

deste ser baseado em assunto e integrado fornece uma visão simples e concisa sobre o negócio. O *DW* é usualmente construído a partir da integração de várias fontes de dados heterogêneas (banco de dados relacionais, arquivos de texto e registros de transações *on-line*). A não volatilidade diz respeito à forma como os dados são tratados *no DW*, neste último eles são carregados e acessados, mas as atualizações geralmente ocorrem no ambiente operacional. A última característica significativa deste tipo de base de dados é ser variável em relação ao tempo, possibilitando a manutenção de uma perspectiva histórica dos dados.

O processo de consolidação das diferentes fontes de dados é realizado por intermédio de técnicas de integração e limpeza, as quais são aplicadas nos dados originais antes do seu armazenamento efetivo no *DW*. Esse processo é denominado de ETC (extração, transformação e carga), e visa adequar os dados originais através de transformações, assegurando que estes sigam padrões desejados, conforme as convenções de nomes, os atributos físicos e as unidades de medidas de atributos [HAN01]. Desta forma, é possível garantir que os dados sejam integrados de forma consistente e com qualidade, possibilitando à tomada de decisão a partir de valores confiáveis.

Tipicamente, um *DW* é construído segundo uma estrutura multidimensional, a qual apresenta como componentes principais: tabelas fato e dimensão. A primeira apresenta atributos numéricos que caracterizam fatos do negócio e atributos chaves que permitem um relacionamento com as dimensões. A segunda representa as diferentes perspectivas ou entidades de análise. Esse tipo de repositório é o principal componente de ambientes de apoio à tomada de decisão; a sua utilização é uma tendência na indústria da informação [HAN01].

2.4 Processo de Descoberta de Conhecimento em Base de Dados

A descoberta de conhecimento em base de dados, também conhecida como *KDD*, é um processo não trivial de identificar padrões válidos, novos e potencialmente úteis em base de dados [FAY96]. Este se caracteriza por ser um processo iterativo e interativo, no qual várias etapas são executadas considerando a decisão do usuário.

Segundo [FAY96] e [HAN01] o processo de *KDD* é composto das seguintes etapas: (i) limpeza e integração (*Cleaning and Integration*); (ii) seleção e transformação (*Selection and Transformation*); (iii) mineração (*Data Mining*); e (iv) avaliação e apresentação (*Evaluation and Presentation*). Já [TAN06] descreve o mesmo processo como sendo constituído de três grandes etapas: pré-processamento, mineração e pós-processamento de dados. Entre as etapas citadas, a

mineração é a mais importante. É nela que algoritmos de descoberta de conhecimento são aplicados sobre os dados buscando encontrar informações úteis.

A Tabela 2 apresenta um comparativo entre as três abordagens citadas, onde as linhas correspondem aos diferentes processos de *KDD* e as colunas representam as suas etapas equivalentes.

Tabela 2: Comparativo entre as diferentes propostas do processo de *KDD*.

Proposta	Etapas		
[TAN06]	Pré-Processamento	Mineração de Dados	Pós-Processamento
[FAY96] e [HAN01]	Limpeza Integração Seleção Transformação	Mineração de Dados	Avaliação Apresentação

O processo de *KDD* proposto por [HAN01] é ilustrado na Figura 2. Um fato que o diferencia das propostas de [FAY96] e [TAN06] é a presença de um *DW* entre as etapas: (i) limpeza e integração e (ii) seleção e transformação. Ele supõe o uso de um *DW* para armazenar os dados relevantes ao problema de forma concisa e organizada, a partir do qual os dados podem ser selecionados e transformados e, então, submetidos ao algoritmo de mineração. Os outros dois autores não prevêem este repositório, admitem apenas que os dados sejam extraídos de suas fontes originais, e por fim preparados para a mineração.

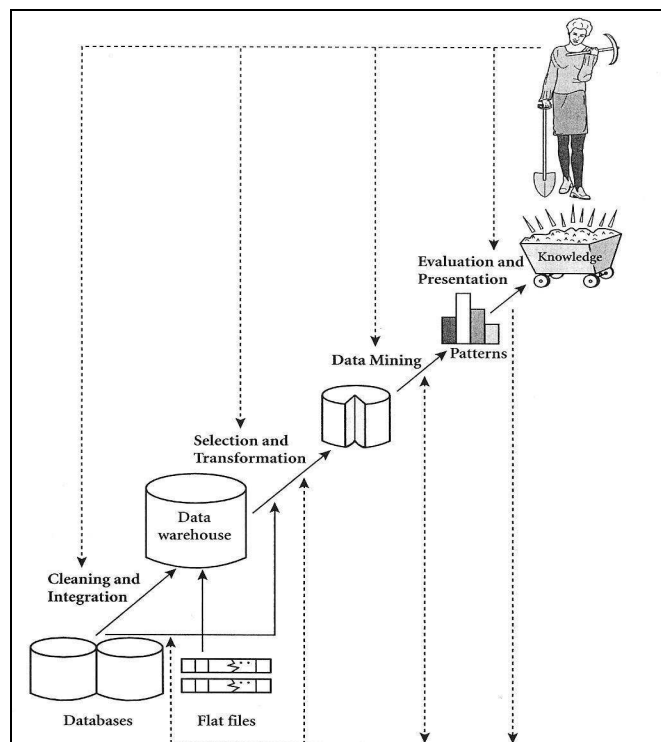


Figura 2: Processo de descoberta de conhecimento (*KDD*) [HAN01].

2.4.1 Pré-processamento de Dados

O pré-processamento ou preparação é a etapa do processo de *KDD* onde os dados são tratados, de forma a se adequarem à entrada do algoritmo de mineração. Esta engloba uma série de outras etapas menores, tais como limpeza, integração, seleção e transformação. Ela é considerada bastante laboriosa e consome em torno de 85% do tempo necessário para executar o processo de *KDD* [LIR07]. No entanto, é através da aplicação de uma adequada preparação que os algoritmos de mineração obtêm resultados satisfatórios. Muitas técnicas voltadas para preparação de dados são sugeridas por [HAN01] e [TAN06]:

- Agregação – busca sumarizar os dados em diferentes perspectivas como, por exemplo, combinando dois ou mais objetos em um único objeto, reduzindo o escopo a ser minerado;
- Amostragem – visa selecionar um determinado subconjunto dos dados que tenham certas características, visando também reduzir o escopo da mineração.
- Redução de Dimensionalidade – busca reduzir o número de atributos (colunas) de um conjunto de dados, através da eliminação de atributos irrelevantes ou redundantes. Esta técnica é um caso particular de seleção de atributos, a qual será apresentada a seguir;
- Seleção de Atributos – assim como a redução de dimensionalidade, essa técnica visa eliminar atributos. Nem sempre os atributos eliminados são irrelevantes, mas o subconjunto selecionado deve ser tão representativo quanto seriam os dados originais. A necessidade de selecionar atributos está associada ao desempenho dos algoritmos de mineração, uma vez que os mesmos apresentam desempenho melhor, em termos de velocidade de execução e interpretabilidade dos modelos, se a dimensão dos dados é menor. A seleção de atributos pode ser realizada de três formas:
 - Experiência sobre os dados: se o usuário conhece bem os dados que está preparando, esse pode decidir quais atributos do conjunto podem ser eliminados.
 - Método *Filter*: consiste na seleção dos atributos por alguma abordagem independente do algoritmo de mineração. Por exemplo, através da construção de uma matriz de correlação de atributos; assim, se dois atributos forem altamente correlacionados um deles pode ser eliminado.
 - Método *Wrapper*: este método usa o próprio algoritmo de mineração para encontrar o melhor subconjunto de atributos.
- Criação de Atributos – permite que, baseado em valores de outros atributos já

- existentes, seja possível criar outro atributo num escopo menor;
- **Categorização** – permite transformar atributos contínuos (numéricos) em atributos discretos (categorias). A necessidade de categorizar atributos surge quando se trabalha com a técnica de classificação, pois esta só permite atributo classe categórico. A transformação de um atributo contínuo em categórico envolve duas etapas: (i) decidir o número de categorias que o atributo terá e, (ii) mapear os valores contínuos para essas categorias. No primeiro passo, após os valores contínuos terem sido ordenados, estes são divididos em n intervalos, determinando $n-1$ pontos de divisão. Num segundo momento, todos os valores pertencentes a um determinado intervalo são mapeados para a mesma categoria. A dificuldade em categorizar atributos está em decidir o número de intervalos e os limites deles. O resultado da categorização é um conjunto intervalos ($[x_0, x_1],]x_1, x_2], \dots,]x_{n-1}, x_n]$).
 - **Transformação de Atributos** – busca aplicar alguma regra que seja inferida sobre os valores de um dado atributo como, por exemplo, normalizar uma escala de valores.

Todas as técnicas descritas são usadas quando os dados são extraídos diretamente a partir de *DW*. Nesta situação os dados já estão previamente adequados para serem armazenados no *DW*, pois estes passam por um processo de ETC, e, assim, precisam apenas ser adequados de acordo com as necessidades do algoritmo de mineração.

Porém, para realizar mineração, muitas vezes apenas os dados disponíveis no *DW* não são suficientes [WIT05]. Desta forma, é necessário recorrer a fontes de dados originais, tais como bases de dados das ferramentas de gestão de projetos. Nessas bases, os dados ainda são brutos; neste caso, para que estes sejam usados na mineração é interessante que os mesmos passem pela etapa de ETC, para que adquiram o mesmo nível de qualidade dos dados armazenados no *DW*.

2.4.2 Mineração de Dados

A mineração de dados é a etapa mais importante do processo de *KDD*. Ela consiste na aplicação de algoritmos para extrair informações úteis e desconhecidas, a partir de grandes repositórios de dados. Segundo [FAY96] e [TAN06] as técnicas de mineração são divididas em duas categorias: descritivas e preditivas.

As descritivas objetivam derivar padrões (correlações, tendências, grupos e anomalias) que sumarizam o entendimento sobre os dados, como exemplo desse tipo de técnica tem-se:

agrupamento, associação e seqüência. Já as preditivas têm a finalidade de prever o valor de um determinado atributo baseado nos valores de outros, exemplos desse tipo de técnica são regressão e classificação.

Neste trabalho é abordada a utilização de classificação como técnica preditiva. Para tanto, a seguir são apresentadas as particularidades desta, bem como os algoritmos e critérios de avaliação dessa técnica.

2.4.2.1 Classificação

A classificação é a tarefa de atribuir objetos a uma entre várias categorias pré-definidas [TAN06]. Através da classificação é possível analisar dados e extrair modelos que descrevem classes ou predizem tendências futuras nos dados [HAN01].

Segundo [TAN06] a classificação é a tarefa de apreender uma função alvo f , a qual mapeia cada atributo de um conjunto X para uma classe pré-definida y . Os atributos do conjunto X são denominados de explanatórios e o atributo y é chamado de atributo classe. A função alvo mencionada é conhecida como modelo de classificação.

A tarefa de classificação é executada em dois passos. No primeiro, uma porção dos dados, denominada de conjunto de treino, é usada para construir o modelo preditivo. Neste conjunto o atributo classe é conhecido para todos os registros. Em seguida, após o modelo ter sido estabelecido, este é testado com um outro conjunto de dados, denominado conjunto de teste. Neste último, os registros têm o atributo classe desconhecido. É através do conjunto de teste que a capacidade de generalização do modelo é avaliada, ou seja, o quanto do total de registros de teste foi previsto corretamente.

Diversas técnicas são sugeridas para estabelecer modelos de classificação, tais como árvore de decisão, redes neurais e redes bayesianas. Este trabalho aborda o uso da classificação através de um algoritmo de árvore de decisão. Na seqüência, são apresentadas as principais características desta técnica de classificação.

2.4.2.2 Classificação por árvore de decisão

A árvore de decisão é uma técnica muito utilizada em problemas de classificação. Entre as principais características que a tornam bastante difundida está a facilidade de interpretação dos modelos gerados.

A árvore de decisão é um gráfico de fluxo em estrutura de árvore. Cada nodo da árvore representa um teste a ser realizado e as arestas definem um caminho para cada resposta desse

teste. Os nodos folha representam as classes. Os algoritmos de aprendizagem desta técnica adotam a abordagem de divisão e conquista. Assim, partindo de um nodo raiz a árvore é construída recursivamente, dividindo o conjunto de treino em subconjuntos, sucessivamente, de acordo com um critério de divisão, até que cada sub-árvore chegue a um nodo folha. O critério de divisão do conjunto é muito importante no processo de construção da árvore, pois determina o próximo nodo da árvore, se será um nodo interno ou folha. Existem vários critérios de divisão em algoritmos de árvore de decisão; porém este assunto não é tratado neste trabalho. Entre os principais algoritmos de árvore de decisão estão [TAN06]: Hunt, ID3, C4.5 (popularmente conhecido como J.48 [WIT05]) e o CART (*Classification And Regression Trees*).

A principal vantagem de utilizar algoritmos de árvore de decisão está relacionada à facilidade de interpretação dos modelos gerados, uma vez que os mesmos podem ser expressos em regras do tipo Se-Então, facilitando o entendimento por parte dos usuários.

2.4.2.3 Critério de Avaliação de Modelos de Classificação

A avaliação dos modelos de classificação resultantes é fundamental para garantir a credibilidade da etapa de mineração. Segundo [HAN01] os modelos preditivos, obtidos a partir de algoritmos de classificação, podem ser avaliados e comparados de acordo com os seguintes critérios:

- **Acurácia da Predição:** este quesito avalia a habilidade do modelo em predizer as classes alvos de novos registros, ou seja, daqueles que não foram usados para gerar o modelo.

$$\text{Acurácia} = (\text{Número de predições corretas} / \text{Total de número de predições})$$

Essa métrica pode ser tabulada em uma matriz, denominada de matriz de confusão. A Tabela 3 apresenta uma matriz de confusão para um problema binário (atributo classe = 1 ou atributo classe = 0). Os verdadeiros positivos (VPo) e os verdadeiros negativos (VN) representam os valores preditos corretamente. Já os falsos positivos (FP) e os falsos negativos (FN) representam os valores preditos erroneamente.

Tabela 3: Matriz de Confusão.

		Valor Predito	
		Classe = 1	Classe = 0
Valor Real	Classe = 1	VPo	FN
	Classe = 0	FP	VN

- Velocidade: avalia o custo computacional envolvido na generalização e uso do modelo.
- Robustez: habilidade do modelo em fazer previsões corretamente perante dados faltantes ou com ruídos.
- Escalabilidade: habilidade de construir um modelo eficiente dado um grande conjunto de dados.
- Interpretabilidade: este item é referente ao nível de entendimento e discernimento fornecidos pelo modelo em relação ao conhecimento descoberto.

Dentre as cinco métricas apresentadas, a acurácia e a interpretabilidade são tomadas como referências para avaliação dos modelos de classificação estabelecidos no capítulo 5, visto que as mesmas revelam o nível de qualidade do modelo resultante e o quanto o mesmo é de fácil interpretação por parte dos usuários.

2.4.3 Pós-processamento de Dados

O pós-processamento compreende as etapas de avaliação e apresentação do conhecimento extraído. Através da avaliação, o usuário deve reconhecer se os padrões extraídos com a mineração representam conhecimento útil para o negócio. Ela deve ser realizada por intermédio de uma das métricas definidas na seção 2.4.2.3. Já a apresentação é a forma como o conhecimento obtido é mostrado para o usuário; [HAN01] sugere as seguintes formas de apresentação: regras, tabelas, gráficos ou árvores de decisão.

3 DESCRIÇÃO DO CENÁRIO

Este capítulo descreve o cenário onde a pesquisa está inserida: como prever métricas de *software* com o uso de mineração de dados. Para tanto, é apresentada: (i) a problemática a ser tratada; (ii) o ambiente *SPDW+*, o qual é tomado como referência para a proposta; (iii) o cenário real da operação parceira e (iv) a caracterização da contribuição. São relatados os aspectos particulares a cada um desses itens, buscando estabelecer como os mesmos se relacionam neste trabalho.

3.1 Problemática

As empresas de TI devem oferecer produtos e serviços, conforme os prazos e custos estabelecidos com seus clientes, para se manterem competitivas no mercado. Por essas razões, estimativas precisas são essenciais para que essas empresas consigam executar um planejamento dos seus projetos o mais próximo possível do efetivamente realizável, além de estabelecer o entendimento quantitativo dos dados de seus processos.

O modelo de maturidade *CMMI* aborda esta necessidade, e define os requisitos para alcançar essas exigências. O *CMMI* nível 2 estabelece a necessidade de um PM para quantificar a qualidade do PDS. Já o nível 3 requer um repositório para armazenar de forma concisa e organizada as métricas organizacionais, permitindo uma visão unificada e comparável entre os diferentes projetos. Desta forma, a contínua manutenção da qualidade dos processos é alcançada através da mensuração dos mesmos. Com base nas métricas definidas e armazenadas em um repositório, é possível manter um histórico da qualidade dos projetos, bem como realizar a análise, predição e monitoração do seu PDS. Contudo, apenas mensurar não é o bastante. As organizações necessitam obter informações de seus processos de forma rápida, para que assim, possam tomar ações corretivas e impedir falhas nos mesmos.

3.1.1 Análise e Monitoração

A análise consiste no entendimento do histórico de métricas através do uso de técnicas estatísticas, gráficos, acompanhamento de indicadores de qualidade e técnicas de mineração. Por exemplo, analisar os indicadores de retrabalho para corrigir defeitos, tentar encontrar as causas destes, e atuar sobre cada uma delas, visando melhorar os processos do projeto.

A monitoração, por sua vez, tem por objetivo oferecer informações do andamento dos projetos em relação ao que foi inicialmente planejado, a partir de um acompanhamento

regular do planejamento e buscando detectar desvios significativos [SEI06]. Esse acompanhamento pode ser efetuado por intermédio de técnicas específicas de monitoração como, por exemplo, *EVA (Earned Value Analysis)* [PMI04].

A monitoração deve oferecer, também, informações momentâneas e atualizadas, que efetivamente auxiliem na tomada de decisão, permitindo que os problemas possam ser detectados tão logo apareçam, e ações corretivas possam ser executadas, no momento certo. Por exemplo, quando uma determinada tarefa encontra-se em atraso e a sua finalização é pré-requisito para o início de outras tarefas, o gestor pode optar por alocar mais recursos nesta tarefa atrasada, ou transferi-la para outro recurso mais qualificado e/ou disponível, para que a mesma não cause impacto nas demais e não ocasione um atraso do prazo de entrega e aumento do orçamento, ou, pelo menos minimizar esse impacto.

3.1.2 Previsão

A previsão também está relacionada com o estudo de dados passados. Porém, ela objetiva a realização de estimativas confiáveis e próximas de valores reais, bem como na detecção de desalinhos de objetivos, além da descoberta da causa raiz de falhas dos produtos (desatenção do colaborador, falha de lógica, falta de conhecimento, etc). Os resultados da previsão podem ser oferecidos através de modelos preditivos, produzidos por técnicas de mineração sobre métricas de *software*. Dessa forma, os gestores podem se beneficiar do histórico de mensurações para melhorar suas estimativas, e conseguir desempenhar um gerenciamento quantitativo de melhor qualidade, o que constitui o principal objetivo do nível 4 do CMMI. Por exemplo, com base nas características de defeitos como: severidade, causa raiz, fase de origem (análise, projeto, implementação, etc) e tamanho do código, é possível estabelecer através de técnicas preditivas o esforço necessário para corrigi-los.

Assim, a predição pode: (i) estabelecer modelos capazes de auxiliar nas estimativas iniciais do projeto; e (ii) apoiar na análise quantitativa de processos. A predição voltada para o estabelecimento de estimativas mais precisas é útil na fase inicial do projeto quando o gerente deve estipular os objetivos iniciais de referência para o trabalho e quando deseja também prever o esforço para corrigir problemas. Já a análise quantitativa mostra-se útil durante o andamento do projeto, para, dada uma situação presente, permitir identificar as chances de ocorrerem desalinhos dos objetivos por, por exemplo, probabilidades. As técnicas de Mineração de dados entram neste contexto como uma abordagem capaz de aumentar a capacidade analítica e preditiva sobre as estimativas e o desempenho quantitativo do PDS,

através da proposta de modelos e técnicas voltadas a este fim.

Contudo, como empregar a mineração de dados para predizer métricas? Isoladamente ou integrada com o repositório organizacional de métricas? É o processo de *KDD* diretamente aplicável ou é conveniente adaptá-lo ao contexto de métricas de *software*? O que precisa ser feito para que resultados satisfatórios possam ser obtidos?

Este trabalho aborda, justamente, o uso de técnica de classificação para predição de métricas de *software*. Para tanto, propõe um processo de *KDD* que incorpora o ambiente de *data warehousing*, denominado *SPDW+*, para previsão de métricas de *software*. Em especial, um método de execução do processo de *KDD* que, em vez de considerar o *DW* como um passo intermediário deste processo, o toma como ponto de referência para a sua execução. Na seção seguinte o *SPDW+* é apresentado, destacando suas funcionalidades e limitações.

3.2 *SPDW+*

O *SPDW+* é um ambiente de *data warehousing* que oferece suporte para análise e monitoração da mensuração da qualidade de *software*, a partir de um Repositório de Dados e de um processo automatizado de ETC das métricas, orientado a serviço [SIL07]. Contudo, não apresenta nenhum recurso de predição. A arquitetura do *SPDW+*, apresentada na Figura 3, está organizada em camada de Integração de Aplicações, de Integração dos Dados, de Apresentação e Repositório de Dados.

A camada de Integração de Aplicações é responsável pela extração automática dos dados diretamente das fontes dos projetos, advindos de diferentes ferramentas utilizadas no ambiente de desenvolvimento de *software*. Os dados são carregados em um componente denominado DSA, que pertence à Camada de Integração de Dados, onde tais dados são devidamente transformados conforme o padrão organizacional. Após o DSA, os dados são carregados para o repositório (*DW*) de maneira incremental, implementado na forma de *web service*. Por último, a Camada de Apresentação permite a exibição dos dados armazenados no *DW*, de acordo com os diferentes perfis de usuário e objetivos de análise.

Na próxima seção é apresentado em detalhes o repositório de dados, o qual é o cerne desta arquitetura. Para tanto, é descrito o modelo analítico e o PM suportados por ele.

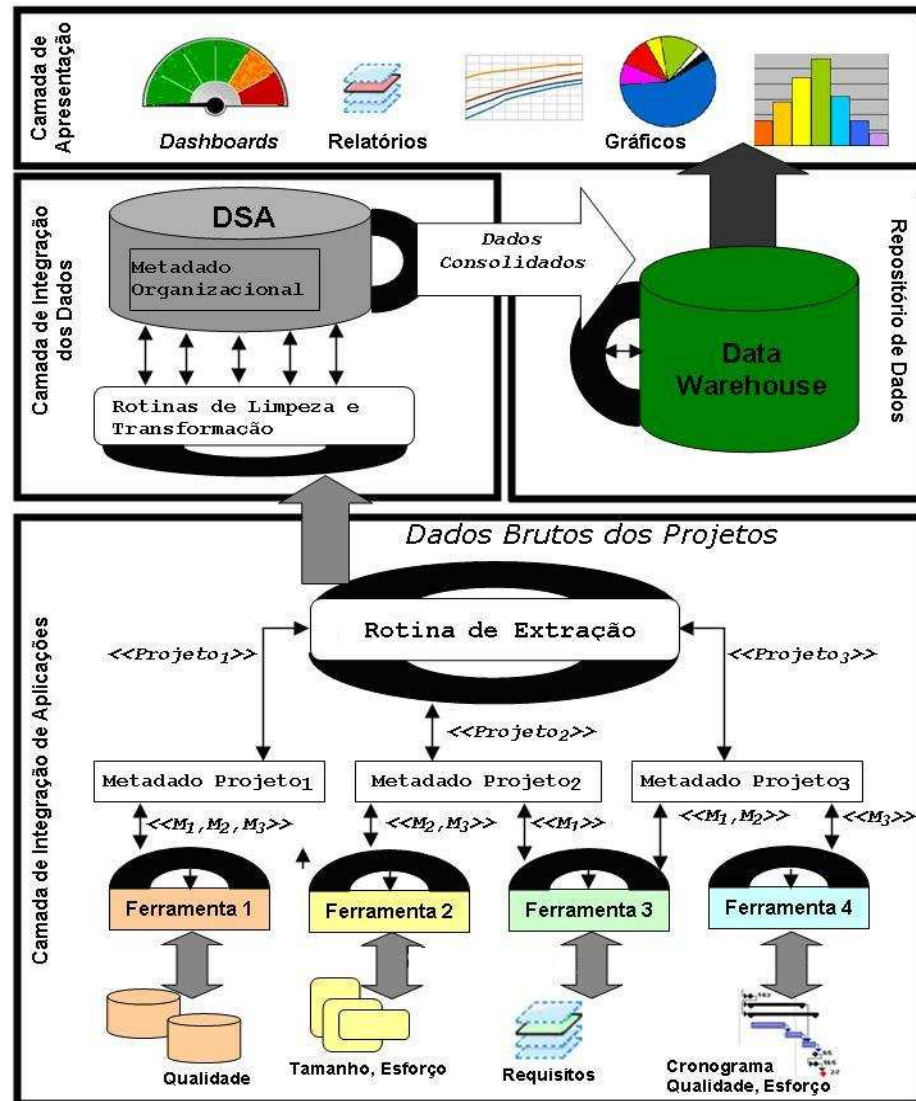


Figura 3: Arquitetura do SPDW+ [SIL07].

3.2.1 Repositório de Dados

O repositório de métricas é construído na forma de um *DW*, definido a partir de um modelo analítico conseqüente dos modelos de processo de desenvolvimento de *software* e do PM. É uma base de dados que armazena, de forma unificada e centralizada, os dados de todos os projetos da organização, permitindo análises e monitorações, em diferentes perspectivas, níveis de sumarizações e papéis organizacionais.

A estrutura analítica do repositório suporta modelos de processo de desenvolvimento tipicamente utilizados por grandes organizações, como os ciclos de vida cascata e iterativo [PRE04]. Desta forma, possibilita que cada projeto contenha uma ou mais versões, sendo que as mesmas podem apresentar um conjunto de iterações ou fases. As fases contêm atividades classificadas segundo seu tipo: *trabalho*, *retrabalho*, *revisão* e *qualidade*. Os defeitos são

mensurados a partir das fases e devem ser classificados conforme o seu grau de severidade (*e.g. alto, médio ou baixo*).

O programa de métricas suportado pelo modelo analítico é apresentado na Tabela 4. Este compreende métricas das seguintes áreas de qualidade: Tempo, Esforço, Tamanho, Custo, Requisitos e Qualidade. As métricas de Tempo são responsáveis por determinar o intervalo de tempo em que uma determinada tarefa deve ser realizada. As de Esforço, Custo e Tamanho apresentam os valores estimados e realizados, em um determinado intervalo de tempo, bem como as variações dos mesmos. A área de qualidade Requisitos engloba as métricas que controlam a variação do número de requisitos inicialmente acordados com os clientes, pois estes podem ser alterados no decorrer do desenvolvimento do produto. E, na de Qualidade tem-se as métricas relativas à qualidade do produto, como número de defeitos internos e externos, eficiência de remoção de defeitos e satisfação do cliente. A monitoração é estabelecida através de métricas específicas (as quais estão destacadas na Tabela 4 por sombreamento), definidas por intermédio da técnica de *EVA* [SIL07].

A Figura 4 ilustra o modelo analítico estabelecido segundo os modelos de PDS e o PM. Ele é organizado segundo um esquema multidimensional do tipo constelação de fato, na qual são definidos dois tipos de tabelas: fato e dimensão. As tabelas fato armazenam métricas nas seguintes granularidades: atividade, versão e defeito. As tabelas dimensão, por sua vez, armazenam os atributos que qualificam os fatos. Esse modelo analítico oferece suporte à análise e monitoração.

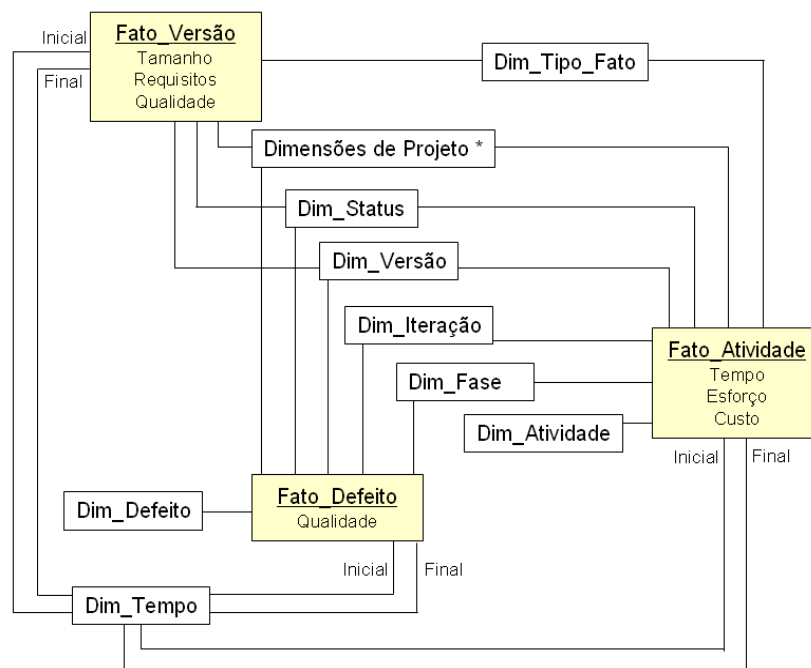


Figura 4: Modelo Analítico do SPDW+ [SIL07].

Tabela 4: Programa de Métricas do SPDW+ [SIL07].

Áreas de Qualidade	Métricas Derivadas	Métricas Diretas
Tempo	VBO – Variação do <i>Baseline</i> Original VBR – Variação do <i>Baseline</i> Revisado	DIR – Data Inicial Real DFR – Data Final Real DIBO – Data Inicial do <i>Baseline</i> Original DFBO – Data Final do <i>Baseline</i> Original DIBR – Data Inicial do <i>Baseline</i> Revisado DFBR – Data Final do <i>Baseline</i> Revisado
Esforço	VEBO – Variação de Esforço do <i>Baseline</i> Original VEBR – Variação de Esforço do <i>Baseline</i> Revisado PR – Produtividade	ER – Esforço Real EBO – Esforço do <i>Baseline</i> Original EBR – Esforço do <i>Baseline</i> Revisado TR – Tamanho Real
Tamanho	VTBO – Variação de Tamanho do <i>Baseline</i> Original VTBR – Variação de Tamanho do <i>Baseline</i> Revisado	TBO – Tamanho do <i>Baseline</i> Original TBR – Tamanho do <i>Baseline</i> Revisado TR – Tamanho Real
Custo	VC – Variação de Custo VCA – Variação de Custo Agregada VPA – Variação de Prazo Agregada IDC – Índice de Desempenho de Custo IDP – Índice de Desempenho de Prazo	CR – Custo Real CBO – Custo do <i>Baseline</i> Original CBR – Custo do <i>Baseline</i> Revisado CRAR – Custo Real da Atividade de Revisão CRFT – Custo Real da Fase de Teste CRAQ – Custo Real das Atividades de Qualidade CRART – Custo Real das Atividades de Retrabalho %TC – % Trabalho Completado DS – Data de <i>Status</i>
Requisitos	VR – Volatilidade de Requisitos	NMA – Nro. de Modificações Aprovadas NMR – Nro. de Modificações Requeridas NRE – Nro. de Requisitos Excluídos NRM – Nro. de Requisitos Modificados
Qualidade	ERD – Eficiência de Remoção de Defeitos DDE – Densidade de Defeitos Entregues DDI – Densidade de Defeitos Internos ERV – Eficiência de Revisão SC – Satisfação do Cliente	NDI – Nro. de Defeitos Internos NDE – Nro. de Defeitos Externos TR – Tamanho Real ISC – Índice de Satisfação do Cliente

3.2.2 Métricas potencialmente úteis para predição

Uma métrica preditiva é aquela que pode ser usada para prever um atributo ou fator de qualidade do PDS. Desta forma, considerando o PM do SPDW+ várias métricas podem ser usadas com essa finalidade, desde que alguma técnica voltada para este fim seja aplicada sobre elas, como por exemplo, técnicas de classificação. Entre as métricas potencialmente úteis para serem preditas tem-se as métricas de Tempo (e.g duração), Esforço (e.g. esforço original), Tamanho (e.g. tamanho original), Custo (e.g. custo original) e Qualidade (e.g. número de defeitos). Todas essas métricas podem ser utilizadas com a finalidade de predição, pois a partir de técnicas preditivas é possível estabelecer o provável valor delas. As métricas de Requisitos, no entanto, são difíceis de serem preditas, pois estas são alteradas ao longo do desenvolvimento dos produtos, de acordo com as necessidades e prioridades do cliente.

3.2.3 Considerações sobre o SPDW+

O SPDW+ apresenta um processo automático de ETC, o qual proporciona a coleta de métricas a partir de diferentes ferramentas e, posterior armazenamento no *DW*. Através deste processo e de métricas específicas, o ambiente oferece suporte à análise e a monitoração. Contudo, o SPDW+ não trata os aspectos relacionados à predição

3.3 Cenário Real Estudado

O cenário abordado neste trabalho é de uma empresa de Operação de *Software*, certificada *CMM3*, que em breve pretende alcançar certificação *CMMI 5*. Esta empresa apresenta um ambiente de *Data warehousing* implementado e operacional, denominado de Base Organizacional (BO), similar ao proposto em [SIL07] que, por sua vez, é uma evolução de [BEC06]. Este ambiente é composto por a) Repositório de Métricas; b) Processo de Carga; e c) Interface de *BI*. A seguir estes três componentes são descritos caracterizando o cenário real do estudo de caso da pesquisa, descrito na Seção 5.

3.3.1 Repositório de Métricas

O repositório de métricas é um *DW*, implementado a partir de um modelo analítico conseqüente do PM e do modelo de PDS da organização. Ele suporta o programa de métricas organizacional, exigência da certificação *CMM3*. A sua estrutura analítica está organizada segundo um esquema multidimensional, do tipo constelação de fatos [BEC06], onde são armazenadas métricas de seis áreas de qualidade (Esforço, Qualidade, Custo, Cronograma, Tamanho e Requisitos) nas seguintes granularidades: versão, fase, iteração, tipo atividade e defeito.

Ele foi desenvolvido para análise; porém as métricas são bastante abrangentes podendo ser úteis para a predição. Apesar de sua abrangência, este apresenta limitações em relação à granularidade das informações e às perspectivas de análise. A partir do mesmo não é possível acompanhar regularmente o desempenho das atividades de maneira individual, ou sumariá-las por intervalo de data, devido à consolidação estabelecida ser por *tipo atividade* e não por *atividade*, pois os projetos classificam suas atividades em: trabalho, retrabalho, revisão e qualidade.

Além disso, as informações de custo são apresentadas, apenas no nível de versões impedindo, por exemplo, a verificação do valor consumido por determinada fase ou iteração, bem como a monitoração do mesmo em uma granularidade menor.

3.3.2 Processo de Carga

O processo de Carga do ambiente é responsável por coletar métricas a partir de cada um dos projetos da organização e consolidá-las no repositório central. Nesse ambiente as métricas de Cronograma (Tempo) encontram-se armazenadas no MS Project. Por sua vez, as de Qualidade podem ser capturadas das ferramentas de acompanhamento de defeitos: *Bugzilla*, *ClearQuest* ou *Mantis*, dependendo do projeto. Já as métricas referentes ao esforço realizado podem estar armazenadas na base de dados do *MS Project*, localizada no ambiente do projeto, ou no Banco de Horas, desenvolvido por um dos projetos da operação. E, por fim, os requisitos e os tamanhos são armazenados em documentos distintos não estruturados (planilhas ou arquivos de texto), variando conforme o projeto.

Desta forma, o processo de carga apresenta uma série de etapas manuais e semi-automatizadas, executadas por pessoas específicas. Em resumo, o processo como um todo é composto pela execução dos seguintes passos: preparação dos dados, homologação dos mesmos, efetivação do processo de carga e homologação dos dados carregados. A preparação tem o objetivo de capturar as métricas de fontes distintas e organizá-las em arquivos estruturados, conforme um padrão determinado pela organização. Após serem coletadas, estas devem passar por uma vistoria para verificação de coerência geral do preenchimento e da formatação, a qual denomina-se homologação das fontes de informação, realizada segundo o documento checklist de carga [HPC05]. Depois que os dados são homologados, eles encontram-se no formato adequado para que possa ser realizado o processo de ETC das métricas. Esse é automatizado por intermédio de pacotes DTS (Data Transformation Services) do SQL Server 2000 [SQL07], tendo como pré-condição que todos os dados dos projetos estejam coletados, transformados e disponibilizados, segundo o padrão organizacional especificado nos documentos de carga. Concluída essa etapa, os dados estão devidamente armazenados no repositório central. Por fim, um novo processo de homologação de dados deve ser realizado, onde ocorre a conferência dos dados apresentados pela Interface de BI.

3.3.3 Interface de *BI*

A interface de *BI* (*Business Intelligence*) utilizada pela organização consiste em um portal web, o qual apresenta as seguintes funcionalidades: (i) recursos *OLAP* (*On-Line Analytical Process*) para acessar os dados (ii) um painel de indicadores; (iii) recursos para visualização das consultas realizadas pelos usuários; e (iv) calendário e integração com correio eletrônico. A interface *OLAP* permite apresentação dos dados através de tabelas dinâmicas. A partir dos dados dessas tabelas, gráficos podem ser gerados e visões sobre os cubos podem ser pré-definidas.

3.3.4 Considerações sobre o cenário real estudado

O ambiente da BO permite extração de dados de diferentes fontes e o armazenamento no repositório central, após uma etapa de transformação. No repositório são armazenadas métricas em diferentes granularidades e áreas de qualidade, maiores detalhes são apresentados em [CUN05]. No entanto, a BO não apresenta recursos nem de monitoração nem de predição, como demonstrado em [SIL07]. As estimativas iniciais do projeto são realizadas empiricamente, baseadas na experiência do gerente de projeto ou através de exaustivas análises sobre os dados do repositório. Para tentar facilitar esse processo, em um dos projetos da organização foi criada uma planilha eletrônica onde são inseridos dados das demandas do cliente. Esta tem sido usada para agilizar o processo de aprazamento de versões e atividades. Porém, os critérios presentes nela para estabelecer a estimativa de esforço estão defasados, causando grandes erros e forçando a correção dos mesmos pela experiência do gerente. Desta forma, a organização tem grande interesse em dispor de recursos ou técnicas que possam dar suporte à predição de seus dados de maneira rápida e o mais confiável possível e, assim, contribuir para o alcance de níveis mais altos de maturidade.

3.4 Caracterização da contribuição

Diante da problemática apresentada, este trabalho tem o objetivo de estabelecer um processo de *KDD*, voltado para o estabelecimento de predição de métricas de *software*. Para tanto, define um método de execução deste processo, denominado de *SPDW-Miner*. Este método estabelece uma série de etapas que guiam a execução do processo de *KDD*. O *SPDW-Miner* incorpora o ambiente de *data warehousing* *SPDW+* e, emprega seu *DW* como ponto de referência para a execução de todo o processo de *KDD*.

Através do conceito de *DW*, apresentado no capítulo 2, é possível identificar a sua representatividade no ambiente organizacional. O fato dele armazenar dados de maneira integrada e baseados em assuntos, mostra que a sua concepção é um tarefa onerosa e complexa, conforme apresentado em [BEC06] e constatado no ambiente da organização parceira [HPC06]. Na passagem de dados de um ambiente operacional (por exemplo, ambiente de um projeto de *software*) para o *DW*, geralmente ocorre uma série de transformações sobre os dados [INM05]. Essas transformações visam adequá-los de acordo com os padrões organizacionais, além de corrigir prováveis inconsistências. Os dados armazenados no *DW* podem ser considerados de alta qualidade e dotados de inteligência a respeito do negócio. O uso de repositórios deste tipo é uma tendência em organizações certificadas *CMM* e *CMMI*, como apresentado em [SUB99], [PAL03], [BEC06] e [SIL07].

O processo de *KDD* proposto é apresentado na Figura 5, ele prevê a interação entre as seguintes entidades: (1) Bases de Dados de Origem e *Data warehousing*, (2) Preparação de Dados, (3) Mineração de Dados, e (4) Avaliação de Resultados. A seguir cada uma delas é detalhada.



Figura 5: Processo de *KDD* voltado para predição de métricas de *software*.

- Bases de Dados de Origem e *Data warehousing*

O *DW* fornece uma visão dos dados disponíveis sobre os projetos, permitindo identificar quais as métricas que podem ser úteis para estabelecer as estimativas. Neste tipo de repositório, o nível de detalhamento das informações auxilia a tomada de decisões estratégicas. Mas, em alguns casos, essa granularidade não é a adequada para a mineração. Com isso, às vezes, torna-se necessário buscar em outras fontes de dados (bases de dados de origem) informações que possam ser agregadas às informações dispostas no *DW* e, desta forma, permitindo dispor de informações relevantes para solucionar o problema de mineração desejado. Na Figura 5, as setas tracejadas entre a entidade Base de Dados de Origem e a Preparação de Dados representam essa possibilidade, caso os dados do *DW* não forem suficientes para compor o arquivo de dados: então são buscadas mais informações nestas bases. Essa busca pode ser efetuada durante a efetivação da preparação. Por exemplo, se em algum momento durante a preparação for percebida a necessidade de mais dados, então existe a possibilidade de retornar tanto ao *DW* quanto às bases de origem. Essa situação é representada pelas setas tracejadas entre a etapa de preparação e o *DW*, e entre a preparação e a Base de Dados de Origem.

- Preparação de Dados

A preparação de dados consiste na aplicação de técnicas específicas para adequar os dados de acordo com as exigências do algoritmo de mineração, tomando por referência a padronização já empregada nos dados dispostos no *DW*. A preparação deve considerar parâmetros já definidos para evitar que um esforço já realizado anteriormente, no processo de ETC, seja repetido sem necessidade. Desta forma, as informações do *DW* servem de parâmetros para a tomada de decisão na etapa de preparação. Ao final da preparação os dados estão prontos para serem minerados.

- Mineração de Dados

A mineração de dados, nesta pesquisa, é utilizada para estabelecer modelos capazes de prever métricas de *software*. Nesse sentido, é usado um algoritmo de classificação para, dadas as características do PDS, definir estimativas, prever desalinhos de objetivos, estabelecer causas de defeitos, entre outras métricas. Para o estabelecimento desses modelos é usada a técnica de classificação (algoritmo J.48), e as razões para a sua escolha encontram-se definidas na seção 2.4.2.2.

- Avaliação de Resultados

Na avaliação os resultados do algoritmo são verificados segundo critérios previamente estabelecidos, tais como: acurácia, taxa de erro e interpretabilidade. Após a interpretação, o usuário pode decidir se o modelo foi satisfatório ou não, podendo retornar à etapa de preparação para realizar mais alguns ajustes ou aplicar alguma técnica para otimizar o resultado obtido. A avaliação é realizada por intermédio da verificação de algum critério, sendo que o mais usual e recomendado na literatura [TAN06] é a acurácia. No entanto, a interpretabilidade do modelo obtido é um critério bastante relevante para o usuário, sendo que este apenas se beneficiará do conhecimento extraído se for capaz de interpretá-lo. Se os resultados obtidos não forem satisfatórios, então o usuário pode retornar à etapa de preparação na tentativa de melhorar a qualidade dos dados, ou ainda, aplicar alguma outra técnica de transformação (essa situação é mostrada através da seta tracejada entre a etapa de Avaliação de Resultados e Preparação de Dados).

3.5 Considerações sobre a descrição do cenário

Este capítulo descreveu a problemática do cenário, apresentando como exemplo dois ambientes onde ela é constatada, o *SPDW+* e o ambiente real da organização parceira. Para tanto, apresentou particularidades de cada um deles e a contribuição deste trabalho. Esta última foi apresentada através do processo de *KDD* proposto para o estabelecimento de predição de métricas de *software* e, do método que estabelece todas as etapas de execução deste processo. No capítulo 4 é apresentado em detalhes o método *SPWD-Miner*.

4 *SPDW-Miner* – Método de Descoberta de Conhecimento

Vários autores de trabalhos relevantes no contexto de mineração de dados, como [FAY96], [HAN01] e [TAN06] abordam e definem o processo de *KDD*, apresentando questões como a importância da preparação de dados, as técnicas voltadas para este fim, algoritmos de mineração e critérios de avaliação dos resultados. Porém, nenhum deles explicita como as etapas do processo devem ser realizadas no nível de detalhe, conforme o abordado em [KIM98] para ETC de um *DW*. O processo de *KDD* proposto por [TAN06] é representado por três grandes etapas: pré-processamento, mineração de dados e pós-processamento. Já o proposto por [FAY96] e [HAN01] apresenta quatro etapas: (i) limpeza e integração; (ii) seleção e transformação; (iii) mineração e (iv) avaliação dos resultados.

[HAN01] considera o *DW* como um recurso intermediário entre as etapas (i) e (ii), onde os dados limpos e integrados são armazenados, para então serem submetidos à etapa (iii). Os outros dois autores, [FAY96] e [TAN06], não mencionam a utilização de repositórios intermediários. Este trabalho sugere um processo de *KDD* diferente, onde o *DW* é ponto de referência para a execução das etapas. A vantagem dessa abordagem está em se beneficiar da qualidade dos dados armazenados no *DW* e da inteligência neles incorporada para guiar a eventual busca por dados adicionais. Outro aspecto relevante neste contexto, é que organizações *CMMI* que necessitam de recursos de predição, geralmente, apresentam repositórios construídos na forma de um *DW*, segundo [SUB99], [PAL03], [BEC06] e [SIL07].

Desta forma, este capítulo apresenta o método para realização do processo de *KDD*, denominado *SPDW-Miner*, o qual tem por objetivo auxiliar usuários de métricas de *software* na obtenção de modelos preditivos. Para tanto, apresenta-se uma seqüência de etapas que, se realizadas coerentemente, podem contribuir para a obtenção de bons resultados na etapa de mineração. A seguir, é definido o perfil do usuário que se beneficiará com o método, as etapas que o constituem e como estas devem ser realizadas.

4.1 Perfil do Usuário

O potencial usuário capaz de se beneficiar do método é um profissional da área de qualidade de uma organização de *software*, o qual possui conhecimentos básicos em *BI* e métricas de *software*. E utiliza a mineração de dados para predizer informações, as quais são úteis para auxiliar na tomada de decisão sobre o projeto. Um exemplo de utilização destas é a

realização de estimativas de esforço para correção de defeitos, com maior precisão. A adoção de um perfil conveniente influencia o grau de detalhamento na descrição do método.

4.2 Etapas do Método *SPDW-Miner*

De acordo com [JUN04], um método é uma ferramenta para a aquisição e construção do conhecimento. Este consiste em um conjunto de etapas ordenadamente dispostas a serem executadas, e que tem por finalidade a investigação de fenômenos naturais para a obtenção de conhecimento. E este deve ser objetivo e sistemático para os resultados serem passíveis de reprodução e confirmação.

A seguir são definidas as etapas do método *SPDW-Miner*. No decorrer da definição das mesmas são discutidas as particularidades inerentes ao contexto dos dados e os objetivos a serem alcançados em cada uma delas. Os dados utilizados na execução do *SPDW-Miner* são métricas de *software*, definidas em um programa de métricas padrão de uma operação de *software* de médio ou grande porte. Para propor o método tomou-se como referência o modelo já mencionado em [TAN06] para, então, expandi-lo em etapas. As etapas que compõem o método são:

- A. Estabelecer o objetivo de mineração;
- B. Conhecer os dados disponíveis;
- C. Extrair os dados;
- D. Preparar os dados;
- E. Adequar os dados para o formato de entrada do algoritmo de mineração;
- F. Aplicar o algoritmo de mineração;
- G. Verificar e Interpretar os resultados;
- H. Otimizar o Modelo Resultante.

As etapas A, B, C, D e E do *SPDW-Miner* equivalem à etapa de pré-processamento proposta por [TAN06]. A etapa F corresponde à mineração. Já as etapas G e H correspondem à etapa de pós-processamento. A Figura 6 ilustra as oito etapas de execução do *SPDW Miner*, as quais compreendem todo o processo de *KDD*, conforme segue. Primeiramente é definido o objetivo da mineração (etapa A), o qual deve ser atingido ao término do processo de *KDD*. Logo após, na etapa B, o usuário se familiariza com os dados disponíveis no *DW*, e avalia se há, dentre estes, atributos que sirvam para a execução da mineração anteriormente planejada. Caso os dados disponíveis não sejam suficientes para tal, buscam-se outras bases de dados,

dentro da organização, que possam suprir a carência de informações. Conhecidos os dados e suas respectivas fontes, o usuário deve extrair-los e consolidá-los em arquivo apropriado (etapa C). Então, na etapa D, devem ser realizadas uma série de transformações visando adequá-los e corrigi-los para serem usados por algoritmos de mineração. A etapa E visa formatar os dados de acordo com as exigências da ferramenta de mineração a ser usada. Na etapa F os dados são minerados, e na etapa G os resultados obtidos são avaliados. É na etapa G que o modelo é verificado para identificar se o mesmo é satisfatório ou não. Se este não foi satisfatório o usuário pode decidir por retornar a uma das etapas anteriores, visando: (i) alterar o objetivos de mineração ou categorização do atributo classe (etapa A); (ii) buscar mais informações relevantes (etapa B ou C); (iii) realizar mais algum ajuste nos dados, por exemplo, testar uma nova categorização (etapa D); (iv) realizar alguma formatação exigida pela ferramenta (etapa E); (v) testar outra configuração do algoritmo ou até mesmo outro algoritmo (etapa F); ou (vi) otimizar o modelo resultante (etapa H). A etapa H é realizada apenas como uma tentativa de otimizar os modelos preditivos estabelecidos, não sendo obrigatória a sua execução. Contudo, se ela é realizada, é necessário retornar à etapa G, para então avaliar o modelo após a otimização. Vale ressaltar que, antes de efetivamente realizar a mineração, o usuário pode retornar a etapas anteriores sempre que necessário. Na seqüência, cada uma das oito etapas são detalhadas.

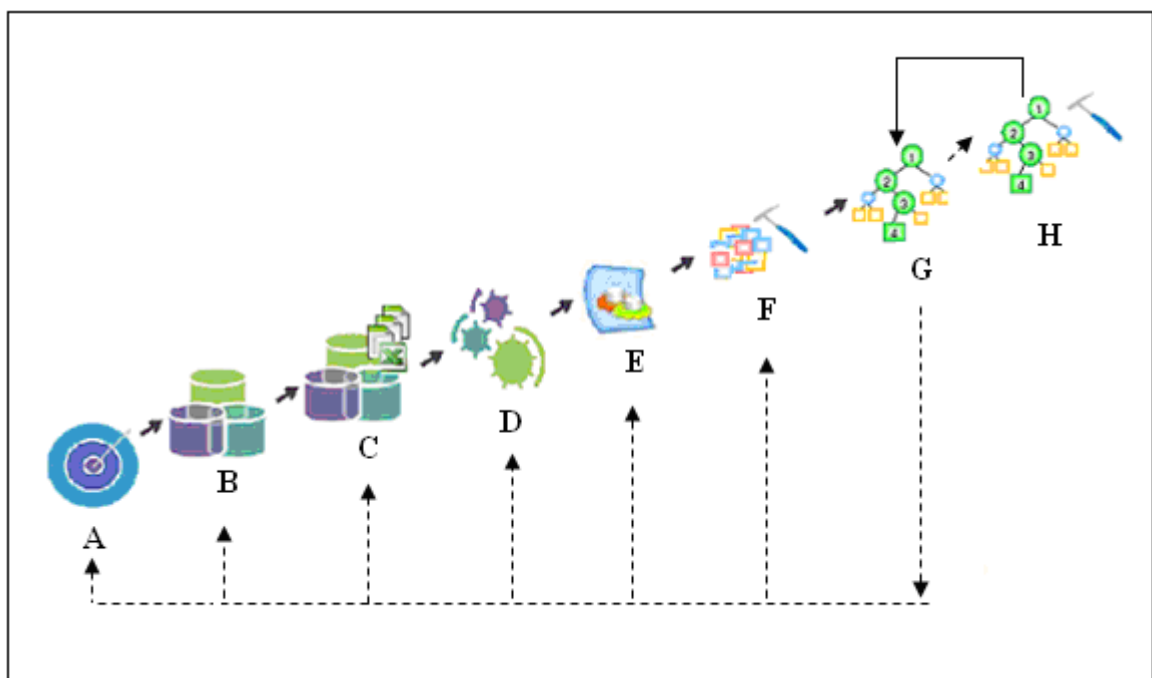


Figura 6: Etapas do *SPDW-Miner*.

A. Estabelecer o objetivo de mineração – quando se deseja minerar dados, o primeiro passo é estabelecer os objetivos de mineração. O objetivo de mineração representa o conhecimento que o usuário pretende extrair a partir dos dados disponíveis. A seguir é apresentada a seqüência de ações que devem ser realizadas nesta etapa:

1. Definir o objetivo de mineração: o usuário deve especificar o conhecimento que pretende prever a partir da mineração. Por exemplo, um objetivo de mineração bastante interessante no contexto de métricas de *software* é estabelecer estimativas de esforço para correção de defeitos;
2. Identificar o atributo classe: de acordo com o contexto dos dados e objetivo de mineração, deve ser definido o atributo classe. No exemplo de prever esforço para correção de defeitos, o atributo esforço (de retrabalho) é a classe;
3. Definir o domínio do atributo classe: quantas e quais são as categorias do atributo classe. Por exemplo, a classe Esforço pode ter três categorias: ‘]-inf - 1 Hora]’, ‘]1-2 Horas]’, ‘> 2 Horas’;
4. Estabelecer os critérios de aceitação dos modelos preditivos resultantes: entre os vários critérios de avaliação de modelos preditivos, já discutidos na seção 2.4.2.3, o usuário deve definir qual métrica usar para avaliar os modelos e quais os limites de valores aceitáveis desta. Este método sugere o uso da métrica de acurácia, pela sua facilidade de interpretação [TAN06]. Outra métrica que deve ser considerada na avaliação dos resultados é o nível de facilidade de interpretação dos modelos resultantes, sendo este caracterizado pelo número de nodos ou níveis da árvore. Pode se ter, como exemplo, uma árvore de decisão que possua uma acurácia $\geq 75\%$ e tenha no máximo 50 nodos, e esta pode ser facilmente interpretada pelo usuário.

No final desta etapa o usuário deve ter estabelecido o objetivo de mineração, o atributo classe, o domínio deste e os critérios de aceitação dos modelos. Após, na etapa B, o usuário deve identificar quais dados são relevantes, os quais podem ser usados como atributos explanatórios.

B. Conhecer os dados disponíveis – Diante da variedade de fontes de armazenamento, muitas vezes é necessário extrair dados de mais de uma fonte para conseguir reunir todos os dados que são úteis para alcançar os objetivos de mineração. Assim, a etapa de conhecer os dados disponíveis consiste na execução dos seguintes passos:

1. Identificar as métricas disponíveis no *DW*: deve ser realizada a investigação no *DW* a respeito das métricas nele armazenadas e os seus assuntos (pontos de vista aos quais as

métricas estão associadas). Através da análise das suas dimensões deve-se verificar os atributos disponíveis, as granularidades em que eles se encontram, o domínio aos quais estes pertencem e a forma como serão integrados. O *DW* é utilizado como um guia para a busca de outros dados. Por exemplo, para alcançar o objetivo de prever esforço de correção de defeitos é necessário reunir métricas que caracterizem os defeitos, para tanto nem todos os dados estão disponíveis no *DW*. Assim, tomando como referência os dados dispostos nele, é possível buscar mais informações, para serem agregadas a estas para contribuir com o objetivo de mineração, sabendo, de antemão, por quais dimensões buscar.

2. Verificar fontes adicionais de informações: caso o *DW* não tenha todas as métricas interessantes para a mineração, devem ser identificadas, na organização, outras fontes de dados que disponibilizem mais informações, tendo sempre como referência os padrões adotados na construção do *DW*. Por exemplo, no *DW* do *SPDW+* não estão disponíveis as informações de severidade de defeitos e número de desenvolvedores, e estas são métricas interessantes para a mineração. Então, estas devem ser buscadas em outras fontes de dados, como a base de dados do projeto, de maneira consistente com os dados que já estão no *DW*.
3. Conhecer o domínio das métricas: para cada métrica disponível o usuário deve conhecer o tipo de cada atributo (*e.g.* categórico e contínuo) e o domínio dos atributos (*e.g.* o atributo severidade pode ter o domínio {Alta, Baixa}), tendo como referência os domínios adotados na construção do *DW*.

Ao final desta etapa, o usuário deve conhecer o conjunto de métricas disponível e o local onde estas estão armazenadas, além das características de cada métrica, as quais passam a ser os atributos explanatórios da classificação.

C. Extração e consolidação dos dados – se for necessário buscar dados de outras fontes, que não sejam o *DW*, é interessante executar alguns dos passos de transformação sugeridos por [KIM98], para adequá-los ao padrão organizacional, sempre tendo em mente as padronizações adotadas no *DW*. O processo de extração pode ser realizado através de consultas *SQL* ou por exportação dos dados a partir da interface da própria ferramenta de gestão. Após terem sido extraídos, uma forma conveniente de acomodar os dados é em arquivos do tipo CSV (arquivos separados por vírgula). Estes arquivos são de fácil manipulação, podendo ser editados como planilha eletrônica ou como arquivo texto. A série de passos a serem realizados nesta etapa é descrita abaixo:

1. Extrair as métricas: as métricas definidas na etapa B devem ser extraídas. A forma de extração, consultas SQL ou exportação dos dados, depende dos recursos disponíveis na ferramenta ou base de dados;
2. Acomodar os dados extraídos: os dados extraídos devem ser dispostos em arquivo, preferencialmente do tipo CSV. Cada coluna do arquivo representa uma métrica e as linhas são os valores por esta assumidos. Ao longo da descrição das etapas do método esse arquivo é denominado de arquivo de dados;
3. Aplicar transformações segundo [KIM98]: se for necessário extrair dados de outras fontes, que não sejam o *DW*, estes devem ser adaptados segundo os passos de transformação proposto por [KIM98]:
 - Limpar ruído:
 - i. Identificar os registros considerados ruídos;
 - ii. Eliminar esses registros do arquivo de dados.
 - Corrigir dados faltantes
 - i. Identificar cada atributo que possui algum registro faltante;
 - ii. Estabelecer o valor que deve substituir os registros faltantes: nesta situação a literatura, em [HAN01] e [TAN06], sugere as seguintes técnicas: (a) eliminar o registro com dado faltante, (b) substituir o valor faltante por uma constante que o caracterize como tal; (c) substituir pelo valor que mais se repete se for atributo categórico ou pela média dos valores se for atributo contínuo, ou (d) estimar o valor faltante através de técnicas de regressão ou agrupamento. Por exemplo, o atributo Tamanho do Código é contínuo, mas para alguns registros ele é faltante. Desta forma pode-se calcular a média dos valores desse atributo e substituir os registros faltantes pela média.
 - iii. Substituir os registros faltantes: no arquivo de dados, os registros faltantes devem ser substituídos pelo valor definido segundo umas das técnicas sugeridas no passo 2.
 - Formatar os valores de acordo com os padrões pré-estabelecidos (*DW*)
 - i. Identificar o atributo que necessita ser adequado aos padrões organizacionais;
 - ii. Realizar a formatação do atributo de acordo com o padrão do *DW*, se disponível; por exemplo, o atributo Severidade na fonte de origem é

categorizado nos níveis '1', '2' e '3'. Já o padrão organizacional define que este seja categorizado como 'A' (alto), 'M' (médio) e 'B' (baixo).

- Eliminar registros desnecessários
 - i. Identificar registros desnecessários, tais como duplicados;
 - ii. Eliminar esses registros do arquivo de dados.

- Combinar fontes de dados
 - i. Verificar a integridade entre as chaves primárias ou realizar a combinação entre atributos não chaves; ter especial atenção com as dimensões do modelo *OLAP* do *DW*;
 - ii. Integrar os dados de fontes diferentes, por exemplo, o atributo Severidade estava em uma tabela e o atributo Causa Raiz estava numa segunda tabela, assim os dois tiveram de ser combinados em um mesmo arquivo, sem perder a integridade dos registros.

Ao final desta etapa, os dados extraídos estão dispostos em um arquivo para serem manipulados na etapa de preparação, onde são adaptados de acordo com as exigências do algoritmo de mineração.

D. Preparar os dados – essa etapa do método prevê a utilização de várias técnicas sugeridas na literatura, em [HAN01], [WIT05] e [TAN06]. Essas são aplicadas de acordo com as necessidades de preparação impostas pelos algoritmos e ferramentas de mineração. Desta forma, as diferentes técnicas são aplicadas interativamente e iterativamente de acordo com as necessidades. Para tanto, o usuário deve examinar o arquivo de dados para identificar quais técnicas de preparação devem ser aplicadas. Essas últimas são agrupadas em quatro tipos: descartar atributos irrelevantes, transformar dados, selecionar atributos e amostragem. A seguir, são descritos os passos que são realizados para a aplicação de cada uma das técnicas.

1. Descartar atributos irrelevantes: os atributos irrelevantes são aqueles que não contribuem para com a generalização do algoritmo de mineração e devem ser desconsiderados. Este tipo de atributo deve ser identificado pelo usuário, e então eliminado. Geralmente, são identificadores de registros como ID, nome ou qualquer outro atributo que contenha uma informação que identifique um único registro. Em

métricas de *software*, o identificador da versão do produto (*e.g.* versão 01.00) é um exemplo desse tipo de atributo. Assim, os procedimentos para removê-los são:

- i. Identificar tais atributos;
- ii. Remover as colunas correspondentes a eles no arquivo de dados.

2. Transformar dados: a transformação é exigida quando alguma alteração nos dados deve ser realizada para adaptá-los à técnica da mineração. Para cada uma das técnicas definidas no capítulo 2 são especificados os seguintes passos:

- Agregação: usada quando for verificada a necessidade de reduzir o escopo a ser minerado; assim, com esta técnica, dois ou mais registros são agrupados. Os atributos numéricos podem ser agregados através da substituição dos valores pela média ou soma dos mesmos, ou até por normalização. Os atributos categóricos devem ser omitidos ou sumarizados.
 - i. Identificar os atributos a serem agregados;
 - ii. Selecionar o critério de agregação como, por exemplo, as horas de retrabalho de uma versão de *software* que são dispostas por colaborador. Para reduzir o escopo, essas podem ser agregadas através da soma do total de horas.
- Redução de Dimensionalidade: realizada quando se tem à disposição muitos atributos para serem usados como atributos explanatórios.
 - i. Identificar atributos que podem ser eliminados.
 - ii. Construir uma matriz de correlação entre os atributos explanatórios e a classe.
 - iii. Verificar quais atributos explanatórios têm alta correlação entre si, pois esses representam atributos que podem ser considerados redundantes, sendo estes passíveis de eliminação. Fica a critério do usuário qual o atributo que deve permanecer e quais devem ser eliminados.
 - iv. Eliminar registros;
- Criação de atributos: usada quando o usuário identifica no contexto dos dados a necessidade de um novo atributo. A criação do mesmo representa a inserção de mais uma coluna no arquivo CSV.

- i. Verificar a necessidade de um novo atributo;
 - ii. Definir o seu tipo e domínio, por exemplo, atributo categórico com o domínio = ('1', '2', '3');
 - iii. Criar a coluna correspondente a ele no arquivo de dados.
- Categorização: usada quando devem ser identificados os atributos contínuos que necessitam ser tratados pelo algoritmo de mineração como categóricos.
 - i. Identificar o atributo que será categorizado;
 - ii. Definir as categorias nas quais serão enquadrados os valores contínuos. A definição das categorias pode ser realizada pelo usuário ou através do uso de algoritmos de agrupamento;
 - iii. Realizar a substituição do valor contínuo pelo novo valor categórico definido.
- Transformação de atributo: é necessária para alterar algum atributo de acordo com as exigências do algoritmo de mineração.
 - i. Definir o atributo que se deseja transformar;
 - ii. Selecionar a técnica de transformação desejada;
 - iii. Aplicar essa técnica, a qual pode ser normalização ou definição de nova nomenclatura para uma determinada categoria. A definição de nova nomenclatura consiste em redefinir as categorias de um atributo categórico quando este possui um número muito grande destas. Por exemplo, o atributo *Tamanho* pode ser categorizado ('1', '2', '3', '4' e '5') e após o processo de transformação, passar a ter outra categorização: ('pequeno', 'médio' e 'grande').
3. Selecionar atributos: a seleção visa identificar quais atributos são mais correlacionados com o problema, e descartar aqueles que não têm correlação para que não atrapalhem na generalização do algoritmo.
 - i. Identificar a necessidade de selecionar atributos, por exemplo, quando o número de atributos explanatórios é muito grande;
 - ii. Definir a técnica a ser usada;
 - iii. Aplicar a técnica.

4. Amostragem: o usuário pode usar esta técnica para selecionar um subconjunto de registros.
 - i. Identificar subconjunto de registros, por exemplo, selecionar apenas os dados onde o atributo *Tamanho* \leq 1000 Pontos de Função;
 - ii. Selecioná-lo e consolidá-lo em um novo arquivo de dados.

No término da etapa de preparação de dados, estes estarão prontos para serem submetidos ao algoritmo de mineração. Isso não garante que os mesmos posteriormente não necessitem retornar à etapa de preparação para serem novamente ajustados. As etapas do método de preparação são iterativas e interativas, como os passos do processo de *KDD*.

E. Adequar os dados para o formato de entrada do algoritmo de mineração – geralmente as ferramentas de mineração necessitam de formatos próprios de entrada de dados para seus algoritmos, fato que garante que os dados estão tratados e perfeitamente adequados para serem minerados. Desta forma, o usuário deve seguir os seguintes passos:

1. Estabelecer a ferramenta de mineração a ser utilizada;
2. Verificar a formatação que a ferramenta exige: o *Weka*, por exemplo, trabalha primariamente com um formato de entrada de dados denominado *Arff* (*attribute-relation file format*). Este tipo de arquivo tem características particulares de formatação, tais como cabeçalho, posição dos atributos no cabeçalho e no corpo do arquivo, e correta denominação do tipo de cada atributo;
3. Enquadrar os dados ao formato esperado, por exemplo, inserir o cabeçalho e formatar a posição dos atributos no arquivo de dados de acordo com as particularidades da ferramenta.

F. Aplicar o algoritmo de mineração - após os dados terem sido preparados e estarem dispostos no formato de entrada, o próximo passo então é executar o algoritmo de mineração desejado.

1. Selecionar o algoritmo desejado;
2. Ajustar parâmetros iniciais do algoritmo: no caso do algoritmo de árvore de decisão J.48, ajustar se o algoritmo deve realizar a poda da árvore e o tipo de validação do modelo (*e.g.* conjunto de treino e conjunto de teste ou validação cruzada);
3. Executar o algoritmo.

G. Verificar e Interpretar os resultados – a verificação dos modelos deve ser realizada por intermédio do critério de aceitação definido na etapa A. Se o critério for satisfeito, então o modelo deve ser interpretado e entendido pelo usuário, para que então este último possa se beneficiar do conhecimento extraído. Cada algoritmo tem uma forma de representar os seus resultados: por exemplo, os algoritmos de árvore de decisão podem ser avaliados segundo a acurácia e a interpretabilidade da árvore obtida. Se os resultados não forem os esperados, o usuário deve decidir se retorna a alguma etapa anterior para fazer mais algum ajuste nos dados, ou se tenta executar a etapa H (otimização).

1. Verificar o resultado do critério de aceitação: examinar o valor do critério de aceitação definido na etapa A. Se a acurácia foi selecionada como critério de aceitação, por exemplo, o seu valor deverá ser verificado.
2. Identificar se o valor do critério de aceitação está nos limites aceitáveis: comparar o valor do critério com o limite mínimo definido para ele na etapa A. Como exemplo, assume-se que a acurácia foi definida como critério, e os limites definidos para aceitação são: modelos que apresentem acurácia de, no mínimo, 70%. Se o modelo não apresentar um resultado satisfatório, o usuário deve tentar outra forma de preparação (etapa D) dos dados ou tentar melhorar o seu resultado através da etapa H. Se o critério apresentar um valor aceitável, pode-se passar para a interpretação do modelo obtido.
3. Interpretar o modelo resultante: se o valor do critério for o aceitável, então o modelo pode ser interpretado. Os modelos preditivos podem ser representados de diversas formas, tais como regras, tabelas, gráficos e árvores de decisão. No caso desta última, o ideal é que ela não apresente muitas ramificações para facilitar a sua interpretação. Para compreender o conhecimento representado por uma árvore, o usuário deve identificar cada um dos seus nodos, e reconhecer o que cada um deles representa. Então, deve-se percorrer a estrutura da árvore para entender o conhecimento representado.

H. Otimizar o Modelo Resultante – quando o resultado do algoritmo de classificação não é o desejado, uma alternativa que pode garantir a melhora deste é a eliminação dos registros que foram classificados erroneamente. Por exemplo, se a acurácia do modelo obtido for muito baixa em relação ao critério definido, o usuário pode eliminar os registros errôneos e aplicar o algoritmo de mineração novamente. Em [WIT05] é sugerido que este processo de

reaprendizado do algoritmo sobre o novo subconjunto de dados deva ser realizado até os resultados serem os desejados.

1. Retirar do arquivo de dados os registros classificados erroneamente;
2. Retornar à etapa F.

4.3 Consideração sobre o método *SPDW-Miner*

O *SPDW* [BEC06] é um ambiente de *Data Warehousing* para apoiar o Programa de Métricas da *HP EAS* Brasil, desenvolvido pelo projeto de parceria entre o Programa de Pós-Graduação de Ciência da Computação da PUCRS (PPGCC-PUCRS) e a *HP EAS* Brasil, durante o seu processo de certificação *CMM3*.

A Figura 3 ilustra a arquitetura do *SPDW+*, organizada em camadas distintas: Camada de Integração das Aplicações (*Application Integration Component*), Camada de Integração dos Dados (*Data Integration Component*) e Camada dos Componentes de Apresentação (*Presentation Components*).

5 ESTUDO DE CASO

Este capítulo apresenta a aplicabilidade do método, *SPDW-Miner*, usando como cenário a operação de *software* parceira. Para tanto, são apresentados vários experimentos, todos com o mesmo objetivo de mineração, porém, abordando a aplicação de diferentes técnicas de preparação sobre as métricas para mostrar a versatilidade e consistência do método. Para tanto, a seguir são propostos (i) o objetivo dos experimentos; (ii) a apresentação dos mesmos, seguindo a seqüência de execução das etapas do método; (iii) os resultados obtidos e análise dos mesmos; e (iv) considerações sobre o estudo de caso.

5.1 Objetivo dos Experimentos

Os experimentos têm o objetivo de mostrar uma aplicabilidade do método *SPDW-Miner*, e avaliar a sua contribuição com a etapa de mineração do processo de *KDD*. Para tanto, é usado o cenário da organização parceira para testá-lo e, então, averiguar os resultados. O cenário real engloba um programa de métricas amplo e um *DW*, conforme já apresentado na seção 3.3.. Para a experimentação o escopo foi reduzido. Buscou-se trabalhar com estimativas de esforço para correção de defeito (*retrabalho*), por ser este de interesse da empresa parceira e, também, por representar uma necessidade real, já que prever corretamente essa métrica representa manter a credibilidade junto aos clientes.

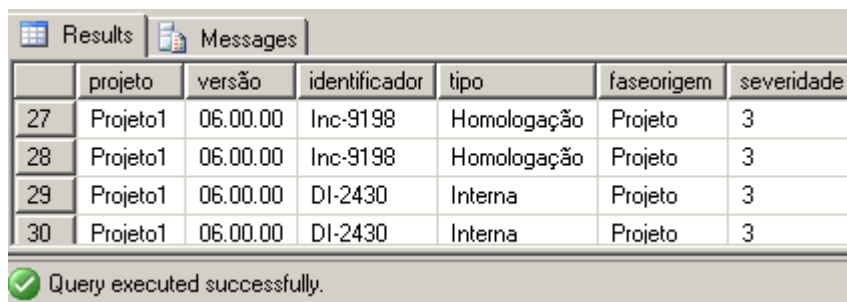
5.2 Experimentos

A seguir é apresentada a seqüência de experimentos executados. Estes foram realizados seguindo as etapas do *SPDW-Miner*. Todos os experimentos seguem o mesmo objetivo de mineração; contudo se diferenciam pelas técnicas de preparação aplicadas. Assim, as etapas A, B, C, E e F do método são comuns a todos os experimentos. As etapas D, G e H são apresentadas na descrição de cada experimento, por apresentarem particularidades para cada um deles.

Na seqüência são definidas as etapas A, B, C, E e F. Posteriormente, são apresentados experimentos e, para cada um, é mostrado como a etapa D foi realizada. A etapa G que consiste na verificação e interpretação dos resultados é apresentada na discussão dos resultados, e a etapa H é apresentada nos experimentos em que ela foi utilizada.

- **A. Estabelecer os objetivos de mineração** – o objetivo de mineração é estabelecer estimativas de esforço para correção de defeito (*retrabalho*). O atributo classe para atingir esse objetivo é a métrica de esforço de *retrabalho*. Inicialmente o domínio do atributo classe foi estabelecido através da categorização em 10 faixas de valores, por intermédio da ferramenta de mineração. Porém, no decorrer da apresentação dos experimentos são apresentadas outras formas de categorização testadas. Para verificar a aceitação dos modelos preditivos foi utilizado o critério da acurácia e interpretabilidade dos mesmos. A acurácia acima de 70 foi estabelecida como aceitável e a interpretabilidade será considerada através do número de nodos da árvore resultante, se a árvore tiver até 50 nodos é considerada interpretável pelo usuário.

- **B. Conhecer os dados disponíveis** – a partir do objetivo de mineração procurou-se no *DW* métricas que fossem interessantes para compor o modelo de estimativa de *retrabalho*. Porém, métricas que fossem de interessantes para compor o modelo de estimativa de *retrabalho*. Porém, percebeu-se que os dados de defeitos estavam muito sumarizados, apresentando as informações de defeitos apenas na granularidade de versão, tais como NDI, NDE, ERD, DDE, DDI e EVR. Essas métricas revelam muitas informações sobre a qualidade de uma versão. No entanto, o *DW* não dispõe de informações sobre o esforço necessário para corrigir um determinado defeito. Conforme já apresentado na seção 3.3.1, nele não constam informações no nível de *Atividade*, apenas na granularidade de *Tipo Atividade*. Assim, não é possível através do mesmo estabelecer o esforço para corrigir um defeito, apenas consegue-se extrair o esforço total de retrabalho despendido em versão do *software*. Desta forma, para atingir o objetivo de mineração estabelecido foi necessário agregar mais informações. Para tanto, buscou-se diretamente na base de dados do *ClearQuest*, ferramenta de acompanhamento de defeitos de um dos projetos da organização. As informações do *DW* serviram como parâmetros para a busca, tais como a informação a ser procurada, o tipo desta e a granularidade da mesma. A base de dados do *ClearQuest* armazena nas suas tabelas, entre outras informações, os seguintes dados: nome do projeto, nome da versão, fase de origem do defeito, severidade, tipo do defeito (interno e externo) e quantidade. A Figura 7 ilustra o resultado de uma consulta contendo: o nome do projeto, a versão, o identificador do defeito, o seu tipo, a fase de origem (fases do ciclo de vida do projeto) e a severidade.



	projeto	versão	identificador	tipo	faseorigem	severidade
27	Projeto1	06.00.00	Inc-9198	Homologação	Projeto	3
28	Projeto1	06.00.00	Inc-9198	Homologação	Projeto	3
29	Projeto1	06.00.00	DI-2430	Interna	Projeto	3
30	Projeto1	06.00.00	DI-2430	Interna	Projeto	3

Query executed successfully.

Figura 7: Resultado de uma consulta na base de dados do ClearQuest.

- **C. Extrair os dados** – os dados da ferramenta de acompanhamento de defeitos foram extraídos com consulta *SQL*. Já os do *DW* foram exportados através da ferramenta de *BI*. Após a extração os dados foram acomodados em um arquivo *CSV*, para então passarem pela etapa de transformação sugerida por [KIM98]. Um trecho desse arquivo bem como uma breve descrição de seu layout encontram-se no Apêndice A. De acordo com as necessidades dos dados as seguintes etapas foram realizadas:

- Corrigir Dados Faltantes: os dados faltantes foram corrigidos através do uso do filtro *ReplaceMissingValues*, disponíveis na ferramenta *Weka*;
- Eliminar Registros Desnecessários: os registros duplicados foram identificados e removidos manualmente do arquivo de dados;
- Combinar Fontes de Dados: a integridade dos dados foi observada, considerando a granularidade do *DW*, assim apenas os dados de mesma granularidade foram extraídos e consolidados no arquivo *CSV*.

- **D. Preparar os Dados** – esta etapa se diferenciou para cada um dos experimentos, e será apresentada na descrição dos mesmos.

- **E. Adequar os dados para o formato de entrada do algoritmo de mineração** – para a mineração foi utilizada a ferramenta *open source*, *Weka 3.5*, desenvolvida pela Universidade de Waikato, na Nova Zelândia [WIT05]. O *Weka* recomenda a inserção de um cabeçalho, onde devem constar todos os atributos explanatórios e o atributo classe com os seus respectivos tipos. Outra recomendação é que todos os registros (colunas) estejam separados por vírgula. O arquivo deve ser salvo com a extensão *.arff*.

- **F. Aplicar o algoritmo de mineração** - no *Weka* é selecionada a técnica de classificação e o algoritmo J.48. Os parâmetros iniciais dos algoritmos foram mantidos padrão

como especificado na ferramenta, ou seja, ativada a função de poda da árvore e usando a validação cruzada.

- **G. Verificar e Interpretar os resultados** – a verificação dos modelos é apresentada na discussão dos resultados dos experimentos.

- **H. Otimizar o Modelo Resultante** – esta etapa, quando utilizada, é apresentada na descrição do experimento.

Nas próximas seções são apresentadas as diferentes configurações dos experimentos realizados. Cada um se caracteriza pela aplicação de diferentes técnicas de preparação, conforme sugeridas na etapa D do *SPDW-Miner*. Para facilitar o entendimento da seqüência de execuções dos mesmos, eles são agrupados de acordo com a técnica de preparação aplicada no atributo classe. Para cada um deles é definido o volume de dados utilizado, os atributos explanatórios, o atributo classe definido, a técnica de preparação aplicada e a acurácia do modelo obtido. No final de cada grupo de experimentos são discutidos os resultados, considerando os critérios definidos na etapa A.

5.2.1 Experimentos com Categorização em 10 faixas de valores

Nos experimentos 1 e 2 é realizada a categorização do atributo classe em 10 faixas de valores. Para tanto, seguiu-se os passos estabelecidos pelo *SPDW-Miner* para categorização. As faixas foram definidas através de um filtro de preparação de dados, denominado de *Discretize*, presente na ferramenta *Weka*. As faixas definidas pelo filtro são: ']-inf-9.322]', '[9.322-18.627]', '[18.627-27.932]', '[27.932-37.237]', '[37.237-46.542]', '[46.542-55.847]', '[55.847-65.152]', '[65.152-74.457]', '[74.457-83.762]', '[83.762-inf['. Esse filtro divide o atributo contínuo em intervalos de mesmo tamanho. A seguir são definidos esses dois experimentos em maiores detalhes.

- **Experimento 1**

- Volume de dados: 9280 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).

- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi realizada apenas a categorização do atributo classe. Este foi categorizado em 10 faixas de valores, são elas: '(-inf-9.322]', '[9.322-18.627]', '[18.627-27.932]', '[27.932-37.237]', '[37.237-46.542]', '[46.542-55.847]', '[55.847-65.152]', '[65.152-74.457]', '[74.457-83.762]', '[83.762-inf['. A Figura 8 apresenta as categorias do atributo classe e a distribuição dos registros em cada uma.
- Acurácia do Modelo: 93,0172%.
- Número de nodos: 64.

Label	Count
'(-inf-9.322]'	8563
'[9.322-18.627]'	584
'[18.627-27.932]'	87
'[27.932-37.237]'	29
'[37.237-46.542]'	8
'[46.542-55.847]'	3
'[55.847-65.152]'	2
'[65.152-74.457]'	2
'[74.457-83.762]'	0
'[83.762-inf['	2

Figura 8: Categorias do atributo classe.

- **Experimento 2**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi realizada uma transformação no atributo *Causa Raiz*, por sugestão de um especialista do domínio. As categorias desse atributo foram renomeadas. Desta forma, foi necessário executar os passos de criação de um novo atributo para adequá-lo à nova nomenclatura. Logo após, a coluna no arquivo de dados que representava a *Causa Raiz* com a antiga nomenclatura foi eliminada. O especialista definiu,

também, que os registros que apresentavam o atributo *Causa Raiz* = *'Inc_corrigida_por_DI'* representavam registros duplicados, logo haviam de ser removidos, restando 8986 registros no arquivo de dados. Essas duas transformações foram mantidas em todos os demais experimentos. O atributo classe foi categorizado em 10 categorias: $(]-\text{inf}-9.322]$, $]9.322-18.627]$, $]18.627-27.932]$, $]27.932-37.237]$, $]37.237-46.542]$, $]46.542-55.847]$, $]55.847-65.152]$, $]65.152-74.457]$, $]74.457-83.762]$, $]83.762-\text{inf}[$.

- Acurácia do Modelo: 93,0825%.
- Número de nodos: 125.

5.2.1.1 Discussão dos resultados com Categorização em 10 faixas de valores

Apesar de a acurácia dos modelos resultantes ser alta, os mesmos não foram satisfatórios. A categorização realizada concentrou um grande número de registros em uma única categoria $(]-\text{inf}-9.322]$), conforme ilustra a Figura 8. Os modelos ficaram tendenciosos, classificando a maioria dos registros nessa faixa de valor. Verificou-se também que os modelos foram construídos em função do atributo *Número de Colaboradores*, como ilustra a Figura 9. Para tentar melhorar os resultados, os próximos experimentos abordam outras formas de categorização do atributo classe e outras técnicas de preparação de dados.

Experimento 1	Experimento 2
Num_Usuarios = 1: $(]-\text{inf}-9.322]$	Num_Usuarios = 1: $(]-\text{inf}-9.367]$
Num_Usuarios = 2: $(]-\text{inf}-9.322]$	Num_Usuarios = 2: $(]-\text{inf}-9.367]$
Num_Usuarios = 3: $(]-\text{inf}-9.322]$	Num_Usuarios = 3: $(]-\text{inf}-9.367]$
Num_Usuarios = 4: $(]-\text{inf}-9.322]$	Num_Usuarios = 4: $(]-\text{inf}-9.367]$
Num_Usuarios = 5: $(]-\text{inf}-9.322]$	Num_Usuarios = 5: $(]-\text{inf}-9.367]$
	Num_Usuarios = 6
	TipoBase = DEF_PRE_REL
	Causa_Raiz = Falta_atenção_envolvido
	Fase_Origem = Infra_Estrutura

Figura 9: Trecho dos modelos preditivos obtidos nos experimentos 1 e 2.

5.2.2 Experimentos com Categorização em 13 faixas de valores

Nos experimentos 3 a 8 o atributo classe foi categorizado através do uso do algoritmo de Agrupamento *K-Means*, disponível no *Weka*. O algoritmo categorizou os dados em 13 faixas distintas: $(]-\text{inf}-0.5]$, $]0.5-1]$, $]1-1.5]$, $]1.5-2]$, $]2-3]$, $]3-4]$, $]4-5]$, $]5-6]$, $]6-$

8]', '[8-10]', '[10-12]', '[12-16]', '>16 Horas'). A seguir são detalhados os experimentos realizados com essa categorização.

- **Experimento 3**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi utilizada a categorização do atributo classe através do uso da técnica de Agrupamento. No *Weka* os dados foram submetidos ao algoritmo de agrupamento *K-Means*. Este estabeleceu as seguintes categorias: (']-inf-0.5]', ']0.5-1]', ']1-1.5]', ']1.5-2]', ']2-3]', ']3-4]', ']4-5]', ']5-6]', ']6-8]', ']8-10]', '[10-12]', '[12-16]', '>16 Horas').
- Acurácia do Modelo: 26.5969 %.
- Número de nodos: 1632.

Com a nova categorização a distribuição de registro por valor do atributo classe ficou mais uniforme. Contudo, a acurácia ficou muito baixa. Através da análise da matriz de confusão, apresentada na Figura 10, percebe-se que os maiores erros estavam em posições próximas da diagonal principal, indicando que o erro do classificador foi em relação à categoria imediatamente inferior ou superior a categoria correta. Por exemplo, na primeira linha da matriz os acertos do classificador são 648, porém na posição ao lado é onde se encontra o maior erro. Este fato indica que a categorização do atributo classe não foi adequada. Outro fato observado é que as regras são sempre estabelecidas em função do atributo *Número de Colaboradores*, como ilustra a Figura 9.

```

=== Confusion Matrix ===
  a  b  c  d  e  f  g  h  i  j  k  l  m  <-- classified as
648 123 37  6 20  0  1  0  0  0  0  0  0 |  a = 0_5
346 343 173 34 126 27  4  0  1  0  0  0  0 |  b = 1
106 229 247 89 273 43 25  3  4  0  0  0  0 |  c = 1_5
 96 135 165 100 337 66 30 10 11  3  2  1  0 |  d = 2
 63 163 184 144 554 172 86 44 29 19  6  5  2 |  e = 3
 30  59  95  69 371 155 91 37 61 17  6 10  2 |  f = 4
  6  31  49  42 227  99 77 43 67 34  6 11  8 |  g = 5
 11  20  15  17 130  85 62 28 52 27 10 13  5 |  h = 6
  8  6  23  24 120  72 67 44 77 44 19 16 24 |  i = 8
  1  10  11  2  56  34 35 26 61 33 15 33 15 |  j = 10
  0  0  3  5  32  9 20 16 41 26 19 16 21 |  k = 12
  0  0  4  2  16 13 12  9 35 34 15 23 34 |  l = 16
  0  0  1  1  6  5  4  4 24 13 17 31 86 |  m = Mais_16Horas

```

Figura 10: Matriz de Confusão do experimento 3.

- **Experimento 4**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi utilizada a categorização do atributo classe através do uso da técnica de Agrupamento, e as categorias estabelecidas foram: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-6]’, ‘]6-8]’, ‘]8-10]’, ‘]10-12]’, ‘]12-16]’, ‘>16 Horas’). Para verificar a influência do atributo *Número de Colaboradores* no modelo resultante do experimento 3, foi realizada uma seleção de atributos, onde este foi retirado do conjunto de atributos explanatórios.
- Acurácia do Modelo: 17.327 %. Através do resultado percebe-se que este atributo é relevante para o modelo.
- Número de nodos: 1228.

- **Experimento 5**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi utilizada a categorização do

atributo classe através do uso da técnica de Agrupamento. As categorias estabelecidas foram: ($]-\infty-0.5]$, $]0.5-1]$, $]1-1.5]$, $]1.5-2]$, $]2-3]$, $]3-4]$, $]4-5]$, $]5-6]$, $]6-8]$, $]8-10]$, $]10-12]$, $]12-16]$, >16 Horas'). O atributo *Número de Colaboradores* sofreu uma transformação, passando a ter uma nova categorização. A Figura 11 apresenta a distribuição de valores para esse atributo. Pode-se perceber que as categorias acima de seis colaboradores têm uma baixa concentração de valores. Desta forma, esses valores foram reorganizados, passando a integrar a nova categoria, *Mais de 5 colaboradores*'. Assim, as categorias do atributo Número de Colaboradores são: ('1', '2', '3', '4', '5', e *Mais_de_5_usuários*'). Para tanto, foi aplicada a técnica de criação de atributo para comportar a nova categorização deste atributo.

- Acurácia do Modelo: 26.4856 %.
- Número de nodos: 1614.

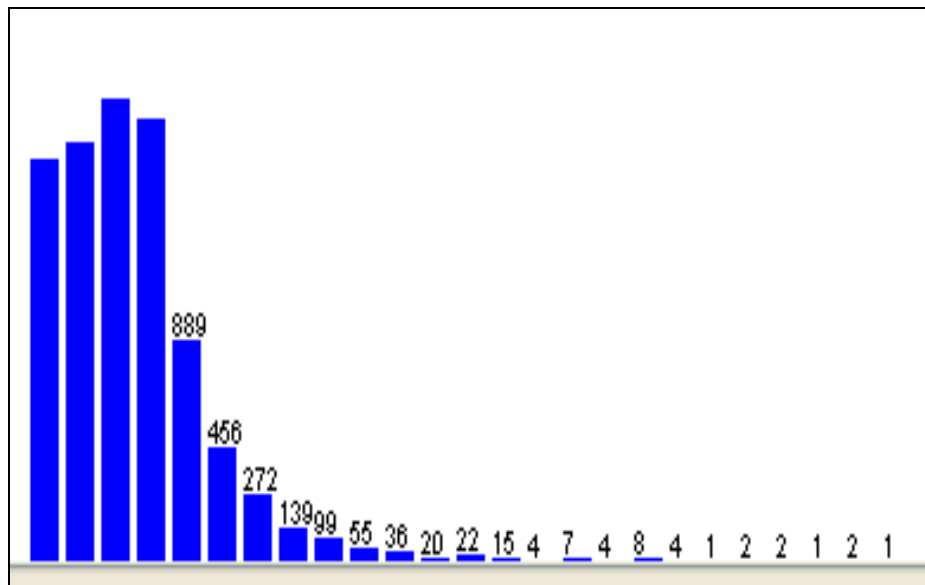


Figura 11: Categorias do Atributo Número de Colaboradores.

- **Experimento 6**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (categórico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).

- Preparação de dados: nesse experimento foi utilizada a categorização do atributo classe através do uso da técnica de agrupamento, as categorias estabelecidas foram: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-6]’, ‘]6-8]’, ‘]8-10]’, ‘]10-12]’, ‘]12-16]’, ‘>16 Horas’). O atributo *Número de Colaboradores* sofreu uma categorização, passando a ter 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’). O atributo *Tamanho*, do tipo contínuo, também foi categorizado. Todos os registros que tinham *Tamanho* = < ‘94.34 PF’ foram enquadrados na categoria ‘P’. E os registros de *Tamanho* > ‘94.34 PF’ na ‘G’. Essa categorização foi adotada levando em consideração a distribuição dos dados, ou seja, tentou-se obter duas categorias uniformes.
- Acurácia do Modelo: 27.4872 %.
- Número de nodos: 440.

- **Experimento 7**

- Volume de dados: 7379 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi utilizada a categorização do atributo classe através do uso da técnica de Agrupamento; as categorias estabelecidas foram: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-6]’, ‘]6-8]’, ‘]8-10]’, ‘]10-12]’, ‘]12-16]’, ‘>16 Horas’). O atributo *Número de Colaboradores* sofreu uma categorização, passando a ter 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’). Visando melhorar a acurácia optou-se por aplicar a técnica de amostragem. Com esta técnica foram selecionados apenas os registros onde o atributo *Tipo de Defeito* = ‘Interno’. Após a amostragem foi realizada a seleção de atributo para eliminar o *Tipo de Defeito*, pois os registros resultantes da amostragem possuem o atributo *Tipo de Defeito* = ‘Interno’.
- Acurácia do Modelo: 26.2908 %.
- Número de nodos: 1329.

- **Experimento 8**

- Volume de dados: 1607 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi utilizada a categorização do atributo classe através do uso da técnica de agrupamento, as categorias estabelecidas foram: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-6]’, ‘]6-8]’, ‘]8-10]’, ‘]10-12]’, ‘]12-16]’, ‘>16 Horas’). O atributo *Número de Colaboradores* sofreu uma categorização, passando a ter 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’). Outra tentativa de preparação de dados foi realizada para melhorar a acurácia: foi aplicada a técnica de amostragem, e então foram selecionados apenas os registros para os quais o atributo *Tipo de Defeito* = ‘Externo’. Após a amostragem foi realizada a seleção de atributos para eliminar o *Tipo de Defeito*, pois os registros resultantes da amostragem possuem o atributo *Tipo de Defeito* = ‘Externo’.
- Acurácia do Modelo: 24.9533 %.
- Número de nodos: 279.

5.2.2.1 Discussão dos resultados com Categorização em 13 faixas de valores

Os resultados dos experimentos mostraram que a categorização através da técnica de Agrupamento não melhorou a acurácia dos modelos. Pode-se perceber que o atributo *Número de Colaboradores* contribui para a melhora da acurácia. A categorização deste último em um número menor de faixas de valores facilitou a interpretação dos resultados. A técnica de amostragem não contribuiu significativamente para a melhora da acurácia. A categorização do atributo *Tamanho* ajudou a melhorar a interpretabilidade do modelo preditivo resultante, porém a acurácia não teve melhora significativa. As Figuras 12, 13 e 14 apresentam trechos dos modelos preditivos obtidos. Percebe-se que o atributo *Número de Colaboradores*, sempre que presente entre os atributos explanatórios, aparece como o atributo raiz da árvore.

Experimento 3	Experimento 4
<pre> Num_Usuarios = 1 TipoBase = DEF_PRE_REL Tamanho <= 126.14: 1 Tamanho > 126.14 Tamanho <= 129.55: 1_5 Tamanho > 129.55 Tamanho <= 184.44: 1 Tamanho > 184.44: 1_5 TipoBase = DI: 0_5 TipoBase = DEF_POS_REL Severidade <= 2: 2 Severidade > 2: 1_5 Num_Usuarios = 2 TipoBase = DEF_PRE_REL Causa_Raiz = Falta_atenção_envolvido Tamanho <= 137.8: 3 </pre>	<pre> TipoBase = DEF_PRE_REL Causa_Raiz = Falta_atenção_envolvido Fase_Origem = Analise_CEF: 3 Fase_Origem = Client Tamanho <= 102.82 Severidade <= 3 Tamanho <= 41.34 Severidade <= 2 Tamanho <= 9.54 </pre>

Figura 12: Trecho dos modelos preditivos obtidos nos experimentos 3 e 4.

Experimento 5	Experimento 6
<pre> Num_Usuarios = 1 TipoBase = DEF_PRE_REL Tamanho <= 126.14: 1 Tamanho > 126.14 Tamanho <= 129.55: 1_5 Tamanho > 129.55 Tamanho <= 184.44: 1 Tamanho > 184.44: 1_5 TipoBase = DI: 0_5 TipoBase = DEF_POS_REL Severidade <= 2: 2 Severidade > 2: 1_5 Num_Usuarios = 2 TipoBase = DEF_PRE_REL Causa_Raiz = Falta_atenção_envolvido Tamanho <= 137.8: 3 </pre>	<pre> Num_Usuarios = 1 TipoBase = DEF_PRE_REL Tamanho = P: 1 Tamanho = G: 1_5 TipoBase = DI: 0_5 TipoBase = DEF_POS_REL Tamanho = P: 0_5 Tamanho = G: 1_5 Num_Usuarios = 2 TipoBase = DEF_PRE_REL </pre>

Figura 13: Trecho dos modelos preditivos obtidos nos experimentos 5 e 6.

Experimento 7	Experimento 8
<pre> Num_Usuarios = 1: 0_5 Num_Usuarios = 2 Fase_Origem = Analise_CEF Tamanho <= 175.96 Severidade <= 1 Tamanho <= 60.42: 1 Tamanho > 60.42: 2 Severidade > 1: 1_5 Tamanho > 175.96 Severidade <= 2: 0_5 Severidade > 2: 1 </pre>	<pre> Num_Usuarios = 1 Tamanho <= 126.14: 1 Tamanho > 126.14 Tamanho <= 129.55: 1_5 Tamanho > 129.55: 1 Num_Usuarios = 2 Causa_Raiz = Falta_atenção_envolvido Tamanho <= 137.8: 3 Tamanho > 137.8 Tamanho <= 182.32: 1 Tamanho > 182.32: 4 </pre>

Figura 14: Trecho dos modelos preditivos obtidos nos experimentos 7 e 8.

5.2.3 Experimentos com Categorização em 9 faixas de valores

Nos experimentos 9 ao 14 optou-se por reduzir o número de categorias do atributo classe na tentativa de melhorar a qualidade dos modelos. A partir da categorização em 13 faixas foram definidas as novas 9 faixas: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-8]’, ‘>8 Horas’). Essas foram estabelecidas tentando manter a uniformidade na distribuição dos valores em cada faixa. A seguir os experimentos são mostrados. Estes seguem a mesma configuração dos experimentos 3 a 8, para que, desta forma, possam ser comparados.

- **Experimento 9**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-8]’, ‘>8 Horas’).
- Acurácia do Modelo: 32.0499 %.
- Número de nodos: 1280.

- **Experimento 10**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-8]’, ‘>8 Horas’). Para verificar a influência do atributo *Número de Colaboradores* no modelo resultante do experimento 9, foi realizada uma seleção de atributo, onde este foi retirado do conjunto de atributos explanatórios.
- Acurácia do Modelo: 18.3063 %.
- Número de nodos: 1106.

- **Experimento 11**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-8]’, ‘>8 Horas’). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’).
- Acurácia do Modelo: 32.1945 %.
- Número de nodos: 1224.

- **Experimento 12**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (categórico), *Número de Colaboradores* (categórico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).

- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: ($]-\infty-0.5]$, $]0.5-1]$, $]1-1.5]$, $]1.5-2]$, $]2-3]$, $]3-4]$, $]4-5]$, $]5-8]$, >8 Horas'). O atributo *Número de Colaboradores* sofreu uma nova categorização, passando a ter 6 categorias: ('1', '2', '3', '4', '5', e '*Mais de 5 colaboradores*'). O atributo *Tamanho*, do tipo contínuo, também foi categorizado. Todos os registros que tinham *Tamanho* \leq '94.34 PF' são enquadrados na categoria 'P'. E os registros de *Tamanho* $>$ '94.34 PF' na 'G'. Essa categorização foi adotada levando em consideração a distribuição dos dados, ou seja, tentou-se obter duas categorias uniformes.
- Acurácia do Modelo: 32.7955 %.
- Número de nodos: 367.

- **Experimento 13**

- Volume de dados: 7379 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: ($]-\infty-0.5]$, $]0.5-1]$, $]1-1.5]$, $]1.5-2]$, $]2-3]$, $]3-4]$, $]4-5]$, $]5-8]$, >8 Horas') O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e '*Mais de 5 colaboradores*'). Visando melhorar a acurácia optou-se por aplicar a técnica de amostragem. Nessa foram selecionados apenas os registros onde o atributo *Tipo de Defeito* = '*Interno*'. Após a amostragem foi realizada a seleção de atributo para eliminar o atributo *Tipo de Defeito*, pois os registros resultantes da amostragem são todos do *Tipo de Defeito* = '*Interno*'.
- Acurácia do Modelo: 30.7223 %.
- Número de nodos: 1081.

- **Experimento 14**

- Volume de dados: 1607 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e

Severidade (numérico).

- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (‘]-inf-0.5]’, ‘]0.5-1]’, ‘]1-1.5]’, ‘]1.5-2]’, ‘]2-3]’, ‘]3-4]’, ‘]4-5]’, ‘]5-8]’, ‘>8 Horas’). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’). Visando melhorar a acurácia optou-se por aplicar a técnica de amostragem. Com essa foram selecionados apenas os registros onde o atributo *Tipo de Defeito* = ‘Externo’. Após a amostragem foi realizada a seleção de atributo para eliminar o atributo *Tipo de Defeito*, pois os registros resultantes da amostragem são todos do *Tipo de Defeito* = ‘Externo’.
- Acurácia do Modelo: 37.15 %.
- Número de nodos: 128.

5.2.3.1 Discussão dos Resultados com Categorização em 9 faixas de valores

Os experimentos 9 a 14 mostraram que a redução do número de categorias do atributo classe contribuiu para a melhoria dos resultados da acurácia, em relação aos experimentos anteriores (1 ao 8). O atributo Número de Usuário mais uma vez se mostrou importante para o resultado. A amostragem não melhorou significativamente os resultados. As Figuras 15, 16 e 17 ilustram trechos dos modelos preditivos obtidos nestes experimentos.

Experimento 9
Num_Usuarios = 1
TipoBase = DEF_PRE_REL
Tamanho <= 126.14: 1
Tamanho > 126.14
Tamanho <= 129.55: 1_5
Tamanho > 129.55
Tamanho <= 184.44: 1
Tamanho > 184.44: 1_5
TipoBase = DI: 0_5
TipoBase = DEF_POS_REL
Severidade <= 2: 2
Severidade > 2: 1_5
Num_Usuarios = 2
TipoBase = DEF_PRE_REL

Figura 15: Trecho do modelo preditivo obtido no experimento 9.

Experimento 10	
TipoBase = DEF_PRE_REL	
Tamanho <= 126.14	
Causa_Raiz = Falta_atenção_envolvido	
Fase_Origem = Analise_CEF: Mais_8Horas	
Fase_Origem = Client	
Severidade <= 3	
Severidade <= 2	
Tamanho <= 5.81: 5	
Tamanho > 5.81	
Tamanho <= 66.78: Mais_8Horas	
Tamanho > 66.78: 8	
Severidade > 2	
Tamanho <= 41.34	
Tamanho <= 16.96	

Figura 16: Trecho do modelo preditivo obtido no experimento 10.

Experimento 11	Experimento 12
Num_Usuarios = 1 TipoBase = DEF_PRE_REL Tamanho <= 126.14: 1 Tamanho > 126.14 Tamanho <= 129.55: 1_5 Tamanho > 129.55 Tamanho <= 184.44: 1 Tamanho > 184.44: 1_5 TipoBase = DI: 0_5 (1224.0/614.0) TipoBase = DEF_POS_REL Severidade <= 2: 2 (2.0/1.0) Severidade > 2: 1_5 (4.0/2.0) Num_Usuarios = 2	Num_Usuarios = 1 TipoBase = DEF_PRE_REL Tamanho = P: 1 Tamanho = G: 1_5 TipoBase = DI: 0_5 TipoBase = DEF_POS_REL Tamanho = P: 0_5 Tamanho = G: 1_5 Num_Usuarios = 2 TipoBase = DEF_PRE_REL

Figura 17: Trecho dos modelos preditivos obtidos nos experimentos 11 e 12.

Experimento 13	Experimento 14
Num_Usuarios = 1: 0_5 Num_Usuarios = 2 Fase_Origem = Analise_CEF Tamanho <= 175.96 Severidade <= 1 Tamanho <= 60.42: 1 Tamanho > 60.42: 2 Severidade > 1: 1_5 Tamanho > 175.96	Num_Usuarios = 1 Tamanho <= 126.14: 1 Tamanho > 126.14 Tamanho <= 129.55: 1_5 Tamanho > 129.55: 1 Num_Usuarios = 2 Causa_Raiz = Falta_atenção_envolvido Tamanho <= 137.8: 3 Tamanho > 137.8

Figura 18: Trecho dos modelos preditivos obtidos nos experimentos 13 e 14.

5.2.4 Experimentos com Categorização em 4 faixas de valores (']inf-2]', ']'2-4]', ']'4-8]', '> 8Horas')

Os experimentos anteriores mostraram que a redução do número de categorias do atributo classe melhorou a acurácia dos modelos. Desta forma, os próximos experimentos são realizados categorizando o atributo classe em apenas quatro faixas de valores (]'inf-2]', ']'2-4]', ']'4-8]', '> 8Horas'). Estas foram escolhidas por representarem intervalos de tempo de interesse da organização parceira, ou seja, um turno de trabalho é representado por 4 horas. A seguir são detalhados os experimentos 15 a 20, usando a categorização definida.

- **Experimento 15**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas de valores: (']inf-2]', ']'2-4]', ']'4-8]', '> 8Horas').
- Acurácia do Modelo: 58.7358 %.
- Número de nodos: 362.

- **Experimento 16**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (']inf-2]', ']'2-4]', ']'4-8]', '> 8Horas'). Para verificar a influência do atributo *Número de Colaboradores* no modelo resultante do experimento 15, foi realizada uma seleção de atributos, onde este foi retirado do conjunto de atributos explanatórios.
- Acurácia do Modelo: 43.3452 %.
- Número de nodos: 85.

- **Experimento 17**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (']inf-2]', '2-4]', '4-8]', '> 8Horas'). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e '*Mais de 5 colaboradores*').
- Acurácia do Modelo: 57.5896 %.
- Número de nodos: 360.

- **Experimento 18**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (categórico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (']inf-2]', '2-4]', '4-8]', '> 8Horas'). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e '*Mais de 5 colaboradores*'). O atributo *Tamanho*, do tipo contínuo, também foi categorizado. Todos os registros que tinham *Tamanho* =< '94.34 PF' são enquadrados na categoria 'P'. E os registros de *Tamanho* > '94.34 PF' na 'G'. Essa categorização foi adotada levando em consideração a distribuição dos dados, ou seja, tentou-se obter duas categorias uniformes.
- Acurácia do Modelo: 57.3448 %.
- Número de nodos: 234.

- **Experimento 19**

- Volume de dados: 7379 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores*

(numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e *Severidade* (numérico).

- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (']inf-2]', ']2-4]', ']4-8]', '> 8Horas'). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e '*Mais de 5 colaboradores*'). Visando melhorar a acurácia optou-se por aplicar a técnica de amostragem. Com essa foram selecionados apenas os registros onde o atributo *Tipo de Defeito* = '*Interno*'. Após a amostragem foi realizada a seleção de atributo para eliminar o atributo *Tipo de Defeito*, pois os registros resultantes da amostragem são todos do *Tipo de Defeito* = '*Interno*'.
- Acurácia do Modelo: 57.2435 %.
- Número de nodos: 335.

- **Experimento 20**

- Volume de dados: 1607 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (']inf-2]', ']2-4]', ']4-8]', '> 8Horas'). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e '*Mais de 5 colaboradores*'). Visando melhorar a acurácia optou-se por aplicar a técnica de amostragem. Com essa foram selecionados apenas os registros onde o atributo *Tipo de Defeito* = '*Externo*'. Após a amostragem foi realizada a seleção de atributo para eliminar o atributo *Tipo de Defeito*, pois os registros resultantes da amostragem são todos do *Tipo de Defeito* = '*Externo*'.
- Acurácia do Modelo: 57.934 %.
- Número de nodos: 27.

5.2.4.1 Discussão dos Resultados com Categorização em 4 faixas de valores ('[inf-2]', '[2-4]', '[4-8]', '> 8Horas')

A redução do número de categorias conseguiu melhorar os resultados tanto em termo de acurácia quanto em interpretabilidade dos modelos preditivos resultantes. Contudo, os mesmos ainda não foram satisfatórios de acordo com os critérios definidos na etapa A do método. As Figuras 19, 20 e 21 mostram trechos dos modelos obtidos.

Experimento 15	Experimento 16
<pre> Num_Usuarios = 1: 2 Num_Usuarios = 2: 2 Num_Usuarios = 3 TipoBase = DEF_PRE_REL Tamanho <= 159 Severidade <= 3: 8 Severidade > 3: 4 Tamanho > 159 </pre>	<pre> TipoBase = DEF_PRE_REL Tamanho <= 118.72 Fase_Origem = Analise_CEF Tamanho <= 48.76: 8 Tamanho > 48.76 Tamanho <= 81.62: 2 Tamanho > 81.62 Tamanho <= 117.66: 8 Tamanho > 117.66: 4 Fase_Origem = Client </pre>

Figura 19: Trecho dos modelos preditivos obtidos nos experimentos 15 e 16.

Experimento 17	Experimento 18
<pre> Num_Usuarios = 1: 2 Num_Usuarios = 2: 2 Num_Usuarios = 3 TipoBase = DEF_PRE_REL Tamanho <= 159 Severidade <= 3: 8 Severidade > 3: 4 Tamanho > 159 Tamanho <= 286.2 </pre>	<pre> Num_Usuarios = 1: 2 Num_Usuarios = 2: 2 Num_Usuarios = 3 TipoBase = DEF_PRE_REL Severidade <= 3: 8 Severidade > 3: 4 TipoBase = DI </pre>

Figura 20: Trecho dos modelos preditivos obtidos nos experimentos 17 e 18.

Experimento 19	Experimento 20
<pre> Num_Usuarios = 1: 2 Num_Usuarios = 2: 2 Num_Usuarios = 3 Fase_Origem = Analise_CEF Causa_Raiz = Falta_atenção_envolvido: 2 Causa_Raiz = Problemas_Colaborador: 2 Causa_Raiz = Especificação_documentação Severidade <= 1 Tamanho <= 118.72: 8 </pre>	<pre> Num_Usuarios = 1: 2 Num_Usuarios = 2 Fase_Origem = Analise_CEF Tamanho <= 78.44: 2 Tamanho > 78.44: 4 Fase_Origem = Client: 2 Fase_Origem = Projeto: 4 Fase_Origem = Server: 4 Fase_Origem = Teste: 4 </pre>

Figura 21: Trecho dos modelos preditivos obtidos nos experimentos 19 e 20.

5.2.5 Experimentos com Categorização em 4 faixas de valores (‘]-inf-1]’, ‘]1-2]’, ‘]2-3]’, ‘> 3 Horas’)

Os experimentos anteriores mostraram um aumento da acurácia dos modelos preditivos, quando era reduzido o número de categorias do atributo classe. Contudo, a acurácia diminuía quando o atributo *Número de Colaboradores* era retirado. Desta forma, os próximos experimentos, de 21 a 24, são realizados considerando o atributo *Número de Colaboradores*, e este categorizado em 6 faixas: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’), e adotando apenas quatro categorias para o atributo classe, as quais são divididas em intervalos de 1 hora (‘]-inf-1]’, ‘]1-2]’, ‘]2-3]’, ‘> 3 Horas’).

- **Experimento 21**
 - Volume de dados: 8986 registros.
 - Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
 - Atributo Classe: *Esforço de Retrabalho* (numérico).
 - Preparação de dados: nesse experimento foi aplicada a técnica de categorização no atributo classe. Esse passou a ter 4 categorias (‘]-inf-1]’, ‘]1-2]’, ‘]2-3]’, ‘> 3 Horas’). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’).
 - Acurácia do Modelo: 58.1349 %.

- Número de nodos: 308.
- **Experimento 22**
 - Volume de dados: 8986 registros
 - Atributos explanatórios: *Tamanho* (categórico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
 - Atributo Classe: *Esforço de Retrabalho* (numérico).
 - Preparação de dados: nesse experimento foi aplicada a técnica de categorização no atributo classe. Esse passou a ter 4 categorias (‘]-inf-1]’, ‘]1-2]’, ‘]2-3]’, ‘> 3 Horas’). O atributo *Número de Colaboradores* sofreu uma categorização, passando a ter 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’). O atributo *Tamanho* também foi categorizado. Esse passou a ser categórico, apresentando as seguintes categorias: ‘P’ e ‘G’. Todos os registros que tinham *Tamanho* =< ‘94.34 PF’ são enquadrados na categoria ‘P’. E os registros de *Tamanho* > ‘94.34 PF’ na ‘G’. Essa categorização foi adotada levando em consideração a distribuição dos dados, ou seja, tentou-se obter duas categorias uniformes.
 - Acurácia do Modelo: 56.8885 %.
 - Número de nodos: 99.
- **Experimento 23**
 - Volume de dados: 7379 registros.
 - Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e *Severidade* (numérico).
 - Atributo Classe: *Esforço de Retrabalho* (numérico).
 - Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (‘]-inf-1]’, ‘]1-2]’, ‘]2-3]’, ‘> 3 Horas’). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’, e ‘Mais de 5 colaboradores’). Visando melhorar a acurácia optou-se por aplicar a técnica de amostragem. Com essa foram selecionados apenas os registros onde o atributo *Tipo de Defeito* = ‘Interno’. Após a amostragem foi realizada a seleção de atributo para eliminar

o atributo *Tipo de Defeito*, pois os registros resultantes da amostragem são todos do *Tipo de Defeito = 'Interno'*.

- Acurácia do Modelo: 55.2378 %.
- Número de nodos: 277.

- **Experimento 24**

- Volume de dados: 1607 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi categorizado nas seguintes faixas: (']-inf-1]', '[1-2]', '[2-3]', '> 3 Horas'). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e '*Mais de 5 colaboradores*'). Visando melhorar a acurácia optou-se por aplicar a técnica de amostragem. Com essa foram selecionados apenas os registros onde o atributo *Tipo de Defeito = 'Externo'*. Após a amostragem foi realizada a seleção de atributo para eliminar o atributo *Tipo de Defeito*, pois os registros resultantes da amostragem são todos do *Tipo de Defeito = 'Externo'*.
- Acurácia do Modelo: 69.8818 %.
- Número de nodos: 26.

5.2.5.1 Discussão dos Resultados com a Categorização em 4 faixas de valores (']-inf-1]', '[1-2]', '[2-3]', '> 3 Horas')

Através da análise da acurácia dos modelos, pode-se verificar que a redução do número de categorias melhorou os resultados. Contudo, os modelos ainda não se enquadram nos critérios definidos na etapa A do método. As Figuras 22 e 23 ilustram trechos dos modelos obtidos.

Experimento 21	Experimento 22
<pre> Num_Usuarios = 1 TipoBase = DEF_PRE_REL Tamanho <= 126.14: 1 Tamanho > 126.14 Tamanho <= 129.55: 2 Tamanho > 129.55: 1 TipoBase = DI: 1 TipoBase = DEF_POS_REL: 2 Num_Usuarios = 2 TipoBase = DEF_PRE_REL Tamanho <= 175.96 </pre>	<pre> Num_Usuarios = 1 TipoBase = DEF_PRE_REL Tamanho = P: 1 (158.0/77.0) Tamanho = G: 2 (230.0/114.0) TipoBase = DI: 1 (1224.0/289.0) TipoBase = DEF_POS_REL: 2 (6.0/3.0) Num_Usuarios = 2 TipoBase = DEF_PRE_REL: Mais_3Horas TipoBase = DI </pre>

Figura 22: Trecho dos modelos preditivos obtidos nos experimentos 21 e 22.

Experimento 23	Experimento 24
<pre> Num_Usuarios = 1: 1 Num_Usuarios = 2 Fase_Origem = Analise_CEF Tamanho <= 227.9: 2 Tamanho > 227.9: 1 Fase_Origem = Client </pre>	<pre> Num_Usuarios = 1 Tamanho <= 126.14: 1 Tamanho > 126.14 Tamanho <= 129.55: 2 Tamanho > 129.55: 1 Num_Usuarios = 2 Tamanho <= 175.96 Tamanho <= 154.76: Mais_3Horas Tamanho > 154.76 Severidade <= 3: 2 Severidade > 3: 1 Tamanho > 175.96: Mais_3Horas Num_Usuarios = 3 </pre>

Figura 23: Trecho dos modelos preditivos obtidos nos experimentos 23 e 24.

5.2.6 Experimentos com Categorização em 2 faixas de valores

Através dos experimentos anteriores percebeu-se que a redução do número de categorias do atributo classe é uma alternativa para melhorar a acurácia dos modelos. Desta forma, nos experimentos a seguir, 25 e 26, é testada a categorização em duas faixas de valores. Utilizou-se a técnica de amostragem para separar o conjunto de registros em duas partes: uma com registros que tinham o esforço de *retrabalho* de até 4 horas e a outra continha os dados com esforço de retrabalho maior que 4 horas. A estratégia de dividir os registros pelo atributo classe se deve à necessidade de tentar encontrar categorias mais precisas, ou seja, reduzir o escopo do problema.

- **Experimento 25**

- Volume de dados: 5897 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi aplicada a técnica de amostragem, onde os registros foram separados em dois subconjuntos. Eles foram divididos segundo o atributo *Esforço de Retrabalho*. Assim, neste experimento foram considerados apenas os registros que possuíam o atributo *Esforço de Retrabalho* \leq '4 horas'. Desta forma, o atributo classe foi categorizado em duas faixas: (']-inf-1.5]', '[1.5 - 4] '). Essas duas foram escolhidas por representarem intervalos bem distribuídos de valores. O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e 'Mais de 5 colaboradores').
- Acurácia do Modelo: 75.0212 %.
- Número de nodos: 17.

- **Experimento 26**

- Volume de dados: 3089 registros
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento foi aplicada a técnica de amostragem, onde os registros foram separados em dois subconjuntos. Eles foram divididos segundo o atributo *Esforço de Retrabalho*. Assim, neste experimento foram considerados apenas os registros que possuíam o atributo *Esforço de Retrabalho* $>$ '4 horas'. Desta forma, categorizou-se o atributo classe duas faixas ('[4-6]', '> 6 Horas'). Essas duas foram escolhidas por representarem intervalos bem distribuídos de valores. O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: ('1', '2', '3', '4', '5', e 'Mais de 5 colaboradores').

- Acurácia do Modelo: 67.2451 %.
- Número de nodos: 24.

5.2.6.1 Discussão dos Resultados com Categorização em 2 faixas de valores

Nos experimentos 25 e 26 foi aplicada a técnica de amostragem, onde os registros foram divididos de acordo com o valor do atributo classe. Os resultados apresentaram modelos preditivos com acurácia de mais de 70%. A Figura 24 ilustra trechos dos modelos obtidos. Porém, com esta abordagem é necessário conhecer de antemão a faixa de esforço a que um determinado registro pertence. No entanto, o objetivo dos experimentos é justamente estabelecer o esforço para correção de defeito. Desta forma, não faz sentido utilizá-la, pois os usuários desconhecem esta informação antecipadamente. A solução adotada foi dividir os registros de acordo com o *Número de Colaboradores*. Essa abordagem foi utilizada, pois o atributo *Número de Colaboradores* tem uma alta correlação com o atributo classe, assim ele foi escolhido para tentar dividir os registros e obter melhor acurácia dos modelos.

Os experimentos 25 e 26 contribuíram para a definição das faixas de valores do atributo classe. Assim, nos próximos experimentos é adotada esta categorização (‘]-inf-1.5]’, ‘]1.5 – 4]’, ‘]4 – 6]’, ‘Mais de 6 Horas’).

Experimento 25	Experimento 26
Num_Usuarios = 1: Ate_1_5	Num_Usuarios_Discretizado = 1:]4_6]
Num_Usuarios = 2	Num_Usuarios_Discretizado = 2:]4_6]
TipoBase = DEF_PRE_REL: De_1_5_ate_4	Num_Usuarios_Discretizado = 3:]4_6]
TipoBase = DI: Ate_1_5	Num_Usuarios_Discretizado = 4
TipoBase = DEF_POS_REL: De_1_5_ate_4	TipoBase = DEF_PRE_REL: Mais_6Horas
Num_Usuarios = 3	TipoBase = DI:]4_6]
TipoBase = DEF_PRE_REL: De_1_5_ate_4	TipoBase = DEF_POS_REL:]4_6]

Figura 24: Trechos dos modelos preditivos obtidos nos experimentos 25 e 26.

5.2.7 Experimentos com Categorização em 4 faixas de valores (‘]-inf-1.5]’, ‘]1.5 – 4]’, ‘]4-6]’, ‘> 6 Horas’)

Para tentar obter modelos mais acurados são testadas as categorias definidas nos experimentos 25 e 26, já que a partir destas obteve-se os resultados mais satisfatórios.

- **Experimento 27**

- Volume de dados: 8986 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento o atributo classe foi novamente categorizado, para tanto adotou-se as categorias definidas nos experimentos 25 e 26: (‘]-inf-1.5]’, ‘]1.5 – 4]’, ‘]4-6]’, ‘> 6 Horas’). O atributo *Número de Colaboradores* passou por uma categorização, onde foram definidas 6 categorias: (‘1’, ‘2’, ‘3’, ‘4’, ‘5’ e ‘Mais de 5 colaboradores’).
- Acurácia do Modelo: 58.7803 %
- Número de nodos: 170.
- Otimização do Modelo Resultante: neste experimento foi aplicada a etapa H do *SPDW-Miner* para otimizar o modelo preditivo. Para tanto, aplicou-se o filtro *RemoveMisclassified* da ferramenta *Weka*, e executou-se o algoritmo novamente. O modelo teve 99.1058% e 160 nodos.

- **Experimento 28**

- Volume de dados: 3300 registros.
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico).
- Preparação de dados: nesse experimento utilizou-se a técnica de amostragem, selecionando apenas os registros que tinham o atributo *Número de Colaboradores* igual a ‘1’ ou a ‘2’ foram usados. O atributo classe foi categorizado, para tanto adotou-se as categorias definidas nos experimentos 25 e 26: (‘]-inf-1.5]’, ‘]1.5 – 4]’, ‘]4-6]’, ‘> 6 Horas’).
- Acurácia do Modelo: 67.8485 %.
- Número de nodos: 15.
- Otimização do Modelo Resultante: neste experimento foi aplicada a etapa H do *SPDW-Miner*, para otimizar o modelo preditivo obtido. Para tanto, aplicou-se o filtro *RemoveMisclassified* da ferramenta *Weka*, e executou-se o algoritmo

novamente. O modelo teve 99.9115 % de acurácia e 17 nodos.

- **Experimento 29**

- Volume de dados: 5897 registros
- Atributos explanatórios: *Tamanho* (numérico), *Número de Colaboradores* (numérico), *Causa Raiz* (categórico), *Fase de Origem* (categórico), *Tipo de Defeito* (categórico) e *Severidade* (numérico).
- Atributo Classe: *Esforço de Retrabalho* (numérico)
- Preparação de dados: nesse experimento utilizou-se a técnica de amostragem, selecionando apenas os registros que tinham o atributo *Número de Colaboradores* igual a '3', '4', '5' e '*Mais de 5 colaboradores*' foram usados. O atributo classe foi categorizado, para tanto, adotou-se as categorias definidas nos experimentos 25 e 26: (']-inf-1.5]', '[1.5 – 4]', '[4-6]', '> 6 Horas').
- Acurácia do Modelo: 53.2982 %.
- Número de nodos: 156.
- Otimização do Modelo Resultante: neste experimento foi aplicada a etapa H do *SPDW-Miner*, para otimizar o modelo preditivo obtido. Para tanto, aplicou-se o filtro *RemoveMisclassified* da ferramenta *Weka*, e executou-se o algoritmo novamente. O modelo teve 98.5084 % de acurácia e 146 nodos.

5.2.7.1 Discussão dos Resultados com Categorização em 4 faixas de valores (']-inf-1.5]', '[1.5 – 4]', '[4-6]', '> 6 Horas')

A amostragem realizada através do atributo *Número de Usuários* não contribui muito com o aumento da acurácia. Para melhorar os resultados foi aplicada a otimização sugerida na etapa H do *SPDW-Miner*. Os modelos resultantes apresentaram valores de acurácia muito satisfatórios e de fácil interpretabilidade (poucos nodos) como ilustrado nas Figura 25, 26 e 27.

Experimento 27	
Num_Usuarios = 1:	1_5Horas
Num_Usuarios = 2	
TipoBase = DEF_PRE_REL:]1_5_a_4Horas]
TipoBase = DI	
Fase_Origem = Analise_CEF:	1_5Horas
Fase_Origem = Client:	1_5Horas
Fase_Origem = Projeto	

Figura 25: Trecho do modelo preditivo obtido no experimento 27 após a otimização.

Experimento 28	
TipoBase = DEF_PRE_REL	
Num_Usuarios = 1:	1_5Horas
Num_Usuarios = 2:]1_5_a_4Horas]
TipoBase = DI	
Num_Usuarios = 1:	1_5Horas
Num_Usuarios = 2	
Fase_Origem = Analise_CEF:	1_5Horas
Fase_Origem = Client:	1_5Horas
Fase_Origem = Projeto	
Severidade <= 1:]1_5_a_4Horas]
Severidade > 1	
Tamanho <= 450.76:	1_5Horas
Tamanho > 450.76:]1_5_a_4Horas]
Fase_Origem = Server:	1_5Horas
Fase_Origem = Teste:	1_5Horas
TipoBase = DEF_POS_REL:	1_5Horas

Figura 26: Modelo preditivo obtido no experimento 28 após a otimização.

Experimento 29	
Num_Usuarios = 3	
TipoBase = DEF_PRE_REL	
Tamanho <= 159	
Severidade <= 3	
Severidade <= 2:]4_a_6Horas]
Severidade > 2	
Tamanho <= 60.42	
Tamanho <= 5.7:]1_5_a_4Horas]
Tamanho > 5.7:]4_a_6Horas]
Tamanho > 60.42	
Tamanho <= 74.2:]1_5_a_4Horas]
Tamanho > 74.2:]4_a_6Horas]
Severidade > 3:]1_5_a_4Horas]
Tamanho > 159	

Figura 27: Trecho do modelo preditivo obtido no experimento 29 após a otimização.

5.3 Considerações sobre o Estudo de Caso

Os experimentos mostraram a abrangência do *SPDW-Miner*, diante da diversidade de cenários apresentados. Várias situações de preparação foram testadas, para mostrar a coerência das etapas do método. O diferencial do processo de *KDD* apresentado pelo *SPDW-Miner* foi comprovado no ambiente real da operação parceira, onde se considerou o *DW* como referência para o processo. Desta forma, a busca por fontes de dados adicionais e a preparação foram parametrizadas pelas informações dispostas no *DW*, facilitando essas duas etapas. Além disso, a execução de um processo de *KDD* organizado e conciso permitiu que resultados satisfatórios fossem alcançados na etapa de mineração, a qual é o principal objetivo deste processo.

6 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos, encontrados na literatura, que são relacionados com o tema de pesquisa. A seguir eles são detalhados e, por fim, é mostrado um comparativo entre eles e a proposta dessa pesquisa.

6.1 NAYAK e QIU [NAY05]

Este trabalho mostra uma aplicação de mineração em métricas de *software*, mostrando como o uso de técnicas de mineração pode contribuir para o melhoramento do processo de desenvolvimento. O objetivo específico deste trabalho é encontrar padrões nos dados, que permitam prever o tempo de reparação de problemas de *software* (esforço de retrabalho) que aparecem durante um PDS, tais como erro de especificação e erro no código.

Os dados utilizados neste estudo são relativos a projetos de *software* de uma companhia de Telecomunicações. Estes dados são coletados em todas as etapas de desenvolvimento de *software*, a partir de relatórios de problemas, por um *software* de acompanhamento de problemas (*bug-tracking*), o qual é mantido na Intranet da organização. Os dados são coletados de todos os departamentos da empresa, totalizando mais de 40.000 PR registrados no sistema.

Para a realização da mineração foram utilizadas as técnicas de classificação e associação. Com a classificação esperava-se encontrar padrões de tempo consumido para corrigir problemas de *software*. Com a associação, pretendia-se descobrir correlações entre os atributos de RP, ou seja, encontrar quais valores de um atributo implicam em determinados valores em outro. Os algoritmos empregados foram: o C5 para a classificação e o CBA para classificação e associação.

Os resultados obtidos não atenderam diretamente o objetivo de mineração proposto pelos autores, que era descobrir o tempo necessário para consertar determinadas classes de problemas. As regras geradas mostraram conhecimento para a classe de problema, e não diretamente para o atributo classe como foi definido no objetivo de negócio. Contudo através da interpretação das regras consegue-se estimar o tempo para conserto.

As taxas de erros dos modelos ultrapassaram os limites previstos. Vários artifícios foram utilizados para tentar minimizá-las, tais como validação cruzada, diferentes tamanhos de conjunto de treinamento, contudo nenhuma destas técnicas trouxe sucesso. Os autores atribuíram esses resultados a alguns ruídos existentes nos dados. Desta forma, novamente é

ressaltada a importância de uma eficiente preparação de dados para garantir sucesso à mineração.

Os padrões encontrados são úteis para auxiliar líderes de projetos a estimar ou prever o tempo necessário para resolver um determinado tipo de problema. E ainda caracterizar certas classes de problemas de *software*. Diante disso, os autores confirmam que a mineração pode contribuir com a Engenharia de *Software*, uma vez que ela permite que conhecimento novo e útil possa ser encontrado para auxiliar no melhoramento da qualidade de um PDS.

6.2 KHOSHGOFTAAR ET AL. [KHO01]

Este trabalho discute como a mineração de dados aplicada à base de dados de Processo de Desenvolvimento de *Software* pode contribuir para a melhoria da qualidade de desenvolvimento de *software* de alto risco. Neste contexto, Khoshgoftaar et al. [KHO01] tentam descrever modelos de qualidade de *software* para sistemas de telecomunicações, que sejam capazes de prever se um determinado módulo de *software* poderá apresentar falha depois de entregue ao cliente. Estes sistemas de telecomunicações requerem alta qualidade para que não apresentem falhas, já que são sistemas complexos e, conseqüentemente, de difícil manutenção.

Os modelos de qualidade são construídos a partir da aplicação de técnicas de classificação nos dados de PDS. Estes modelos devem ser capazes de prever quais módulos poderão apresentar defeitos futuros. Isso permite que esforços possam ser concentrados no desenvolvimento destes módulos, buscando melhorá-los e evitando uma possível falha futura.

Os dados utilizados para realizar a mineração foram extraídos de uma grande base de dados, que contém relatórios de problemas e dados de gestão de configuração de *softwares*. As métricas usadas dizem respeito a atributos do código fonte de um módulo de *software*, tais como relacionamento entre procedimentos do código fonte, número de procedimentos, e número de estruturas de controle, além de informações sobre o tempo gasto para realizar uma alteração no código e o número de alterações feitas por determinada pessoa durante sua carreira (sua experiência profissional).

O pré-processamento dos dados foi considerado pelos autores como indispensável. Na etapa de limpeza dos dados alguns atributos com erro ou com dados faltantes foram eliminados. As transformações foram necessárias, visando melhorar o modelo preditivo resultante. Alguns atributos altamente correlacionados foram transformados, para diminuir o

número de atributos explanatórios. As métricas que tinham média dos valores também foram transformadas, passando a ter o valor total, pois a medida de qualidade era calculada em relação ao total de falhas encontradas pelo cliente; então, os demais atributos também tiveram que considerar valores totais. Foi utilizado o algoritmo CART (*Classification And Regression Trees*). Os resultados dos experimentos foram bastante satisfatórios, mostrando a capacidade da aplicação de mineração de dados em dados de PDS. A taxa de erros dos modelos preditivos foi de apenas de 19,8 %. Através dos resultados, os autores concluíram que modelos preditivos de qualidade podem descrever, competentemente, módulos que possam apresentar defeitos depois de entregues ao cliente.

6.3 NAGAPPAN ET AL. [NAG06]

No estudo de [NAG06] são usadas técnicas preditivas, classificação e regressão, para compor modelos capazes de prever se um módulo de *software* apresentará falhas após a entrega ao cliente. Através dos modelos estabelecidos, os autores, pretendem descobrir se existe um conjunto de métricas que pode ser usado para prever falhar em projetos de *software* distintos. Desta forma, desejam propor uma especificação de como construir, sistematicamente, modelos preditivos de defeitos *post release* (após a entrega ao cliente).

Este trabalho utiliza métricas de cinco produtos de *software* da Microsoft para estabelecer os modelos. Para tanto utiliza características do código fonte dos produtos, tais como número de linhas, complexidade, número de parâmetros de entrada, número de classe, para prever se os mesmos serão propensos a falhas.

Na etapa de preparação dos dados foi construída uma matriz de correlação para identificar quais atributos são altamente correlacionados com o atributo classe. Desta forma, apenas os atributos altamente relacionados foram considerados para a aplicação das técnicas preditivas. Os autores não mencionaram mais nenhuma técnica de preparação que tenha sido aplicada.

Este trabalho relatou que não é possível considerar um conjunto específico de métricas para prever falhas *post release* em todos os projetos. Mostrou, também, que um modelo preditivo, construído a partir de dados de um dado projeto, não consegue estabelecer resultados para dados de outro projeto. Para estabelecer os modelos, os autores descrevem uma especificação de como construí-los. Contudo, a abordagem não é ampla o suficiente, mostrando apenas a sistemática de como selecionar um módulo de *software*, e selecionar o conjunto de métricas a serem usadas para estabelecer os modelos.

6.4 WINCK [WIN07]

Este trabalho propõe um processo de *KDD* para auxílio na reconfiguração de ambientes virtualizados, como o Xen. Este último é um paravirtualizador que permite que várias máquinas virtuais (MV) sejam executadas simultaneamente sobre um mesmo hardware, onde cada uma dessas MV possuem diferentes níveis de recursos.

O processo de *KDD* construído visa melhorar a performance do Xen, verificando qual a melhor alocação de recursos para o mesmo, sugerindo modificações em seus parâmetros. Como fonte de dados são executados diferentes *benchmarks* sobre as MV, a fim de coletar dados referentes ao desempenho das mesmas. Para organizar e armazenar esses dados, foi construído um modelo de *DW*, focalizado em captura de métricas de *benchmarks*, o qual permite que sejam armazenadas quaisquer execuções de benchmarks, em diferentes ambientes computacionais. Sobre os dados devidamente organizados no *DW*, é aplicada mineração de dados, onde são utilizadas tarefas preditivas de classificação, cujo objetivo é que os modelos preditivos gerados sugiram uma configuração vigente, através de novos parâmetros de reconfiguração e, assim, se possa alcançar um ganho de desempenho. Para que a mineração utilizada alcance os resultados esperados, são aplicadas, sistematicamente, técnicas de preparação de dados para a mineração. Essas técnicas buscam trabalhar com os dados já inseridos no *DW*, de maneira que possam ser especialmente úteis para produzir os resultados esperados. Os testes efetuados mostraram a qualidade e abrangência da solução proposta. O trabalho propõe um processo de *KDD* mas não o centra no *DW*, o que é uma característica do *SPDW-Miner*.

6.5 Considerações sobre os trabalhos relacionados

A seguir, na Tabela 5, é apresentada uma comparação entre os trabalhos relacionados, em relação aos aspectos abordados nesta pesquisa. Os aspectos mencionados são os seguintes:

- Contexto abordado: com este item pretende-se conhecer o domínio de conhecimento no qual está sendo aplicada mineração. Por exemplo, neste trabalho será aplicada mineração no contexto de processo de desenvolvimento de *software*.
- Dados Utilizados: especificar que dados foram utilizados como atributos preditivos. Por exemplo, métricas de *software*.
- Objetivo da mineração: estabelecer o conhecimento que se pretende descobrir.
- Pré-processamento: identificar se foi aplicada alguma técnica de preparação.

- Técnica de mineração: qual técnica ou quais técnicas de mineração foram utilizadas.
- Processo de *KDD*: identificar se a proposta define algum método para guiar o usuário na execução do processo de *KDD*.
- *Data warehouse*: identificar se a proposta define a utilização de algum repositório para auxiliar no processo de *KDD*.

Tabela 5: Comparações entre os trabalhos relacionados.

Trabalhos	[NAY05]	[KHO01]	[NAG06]	[WIN07]	[FIG08]
Contexto abordado	PDS	PDS	PDS	Ambientes virtualizados	PDS
Dados Utilizados	Métricas de <i>Software</i>	Métricas de <i>Software</i>	Métricas de <i>Software</i>	Métricas de Benchmarks	Métricas de <i>Software</i>
Objetivo de Mineração	Predizer esforço de Retrabalho	Identificar módulos falhos pós-entrega ao cliente	Identificar módulos falhos pós-entrega ao cliente	Predizer reconfigurações de ambientes virtualizados	Predizer esforço de Retrabalho
Preparação de Dados	Eliminação de ruídos Discretização Transformação	Eliminação de ruídos Seleção de Atributos	Seleção de Atributos	Propostas por [HAN01] e [TAN06]	Propostas por [HAN01] e [TAN06]
Técnica de mineração	Classificação e Associação	Classificação	Classificação e Regressão	Classificação	Classificação
Processo de <i>KDD</i>	Não define	Não define	Define parcialmente	Define	Define
<i>Data warehouse</i>	Não utiliza	Não utiliza	Não utiliza	Utiliza como etapa intermediária do processo de <i>KDD</i>	Utiliza como referência para execução do processo de <i>KDD</i>

Os três primeiros trabalhos relatados, [NAY05], [KHO01] e [NAG06], mostram a aplicação de técnicas de mineração em métricas de *software*. Porém, nesses estudos não fica claro se é adotado um processo de *KDD*, ou se é utilizado um *DW* como parte do processo. Apenas em [NAG06] existe a preocupação de definir uma especificação de como estabelecer modelos preditivos de métricas de *software*. Contudo a especificação apresentada não contempla todas as etapas de um processo de *KDD*, apenas se preocupa em definir as métricas que devem ser consideradas para estabelecer os modelos preditivos. A proposta mais próxima deste trabalho foi a de [WIN07] que estabelece um processo de *KDD* completo para auxiliar na reconfiguração de ambientes virtualizados. Contudo, o mesmo não centra a preparação dos dados no modelo e conteúdo do *DW*, aspecto que, para PDS, mostra-se bastante conveniente.

7 CONSIDERAÇÕES FINAIS

Este trabalho propõe um método para a execução do processo de *KDD*, denominado de *SPDW-Miner*, voltado para o estabelecimento de predições de métricas de *software*, por exemplo: esforço de retrabalho, custo, esforço de trabalho, tamanho. O método é composto por uma série de etapas que guiam os usuários para o desenvolvimento de todo o processo de *KDD*, tomando como referência um repositório de métricas de *software* estruturado na forma de um *DW*. Foram especificadas todas as etapas que compõem o processo de *KDD*, desde o estabelecimento do objetivo de mineração; a extração e preparação dos dados; a mineração até a otimização dos resultados.

Para caracterizar um cenário real de aplicação desta pesquisa foi estudado o ambiente de uma operação de desenvolvimento de *software*, certificada *CMM3*, e uma proposta de evolução do mesmo [SIL07], relatados no capítulo 3. A partir desse estudo foi possível constatar as limitações e necessidades do cenário, onde foi verificada a necessidade da presença de recursos de predição, que possibilitem estimativas mais precisas, as quais podem ser consideradas essenciais para a obtenção de níveis de maturidade mais altos.

A validação da solução proposta foi realizada através da aplicação das etapas do *SPDW-Miner* no contexto da operação parceira. Para tanto, foi definido um objetivo de mineração de interesse da parceira, e então aplicou-se exaustivamente o método. O objetivo da mineração é estabelecer modelos capazes de prever o esforço de retrabalho. Na experimentação foram testadas várias situações de preparação de dados. Desta forma, pode-se constatar a abrangência do *SPDW-Miner*, pois este conseguiu guiar as várias problemáticas constatadas e, por fim, estabelecer resultados satisfatórios na mineração.

O *SPDW-Miner* representa uma inovação em relação aos trabalhos relacionados, através da sua proposta de adotar toda uma sistemática para a execução coerente do processo de *KDD* e, também, por se beneficiar das informações do *DW* para guiar o processo.

7.1 Trabalhos Futuros

A continuidade deste trabalho visa estender os benefícios oferecidos pelo método *SPDW-Miner*. Desta forma, pretende-se atingir os seguintes objetivos:

- Utilizar os modelos preditivos resultantes do processo de *KDD*, guiado pelo *SPDW-Miner*, no ambiente da operação parceira.
- Explorar o *SPDW-Miner* com outras técnicas preditivas.

- Aplicar o *SPDW-Miner* em outros contextos que necessitem de recursos de predição.

REFERÊNCIAS

- [BEC06] BECKER, K.; RUIZ, D.; NOVELLO, T.; CUNHA, V. *SPDW: a Software Development Process Performance Data Warehousing Environment*. In: *Software Engineering Workshop (SEW'06)*, 30, 2006, Bethesda, MD. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2006, p. 107-118.
- [CUN05] CUNHA, V. **Uma Abordagem Orientada a Serviços para Captura de Métricas de Processo de Desenvolvimento de Software**. 2005. 117 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2005.
- [DIC04] DICK, S.; et al. Data mining in *software* metrics databases. **Fuzzy Sets and Systems Journal**, Amsterdam, v. 145, n. 1, p. 81-110, Jul. 2004.
- [FAY96] FAYYAD, U.; PIATETSKY-SHAPIRO G.; SMYTH P. The *KDD* process for extracting useful knowledge from volumes of data. **Communications of the ACM**, New York, v. 39, n. 11, p. 27-34, Nov. 1996.
- [GOP02] GOPAL, A.; et al. Measurement Programs in *Software Development: Determinants of Success*. **IEEE Transactions on Software Engineering**, Piscataway, v. 28, n. 9, p.863-875, Sept. 2002.
- [HAN01] HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. San Francisco: Morgan Kaufmann, 2001. 550 p.
- [HPC05] HP - Hewlett-Packard Company Brasil Ltda. **Checklist de Carga 1.0**. Porto Alegre: HP EAS Brasil, 2005. 15 p. (Relatório Técnico)
- [HPC06] HP - Hewlett-Packard Company Brasil Ltda. **M1 – Um Diagnóstico da Base Organizacional HP EAS Brasil 1.0**. Porto Alegre: HP EAS Brasil, 2006. 23 p. (Relatório Técnico)
- [IEE98] ANSI/IEEE Std 1061-1998. **IEEE Standard for a Software Quality Metrics Methodology**, Piscataway, NJ: IEEE Standards Dept., 1998. 26 p.
- [INM05] INMON, W. **Building the Data Warehouse**. Indianapolis, IN: John Wiley & Sons, Inc, 2005. 543 p.
- [JUN04] JUNG, C. F. **Metodologia para Pesquisa & Desenvolvimento**. Rio de Janeiro: Axcel Books do Brasil, 2004. 312 p.
- [KAN03] KAN, S. **Metrics and Models in Software Quality Engineering**. Boston: Addison-Wesley, 2003. 528 p.
- [KHO01] KHOSHGOFTAAR, M.; ALLEN, E.; JONES, W.; HUDEPOHL, J. Data Mining of Software Development Databases. **Software Quality Journal**, New York, v. 9, n. 3, p. 161-176, Nov. 2001.

- [KIM98] KIMBALL, R. **Data Warehouse Toolkit**. New York, NY: John Wiley & Sons, Inc, 1998. 771 p.
- [LIR07] LI, T.; RUAN, D. An extended process model of knowledge discovery in databases. **Enterprise Information Management**, Bingley, v. 20, n. 2, p.169-177, 2007.
- [NAG06] NAGAPPAN, N.; BALL, T.; ZELLER, A. Mining metrics to predict component failures. In: International conference on Software Engineering, 28, 2006, Shanghai, China. **Proceedings...** New York: ACM, 2006, p. 452-461.
- [NAY05] NAYAK, R., QUI, T. A data mining application: Analysis of problems occurring during a software project development process. **International Journal of Software and Knowledge Engineering**. Brisbane, v.15, n.4, p.647-663, Aug. 2005.
- [PAL03] PALZA, E.; FUHRMAN, C.; ABRAN, A. Establishing a Generic and Multidimensional Measurement Repository in CMMI context. In: Annual IEEE/NASA Software Engineering Workshop, 28, 2003, Greenbelt, MD, USA. **Proceedings...** Los Alamitos: IEEE Computer Society Press, 2003, p.12-20.
- [PMI04] Project Management Institute. **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**. 3rd Edition. Newton Square: Project Management Institute, 2004. 380 p.
- [PRE04] PRESSMAN, R. **Software Engineering**. New York: McGraw-Hill, 2004. 888 p.
- [SEI06] SEI - Software Engineering Institute. **CMMI for Development, Version 1.2**. Pittsburgh: Carnegie Mellon University and Software Engineering Institute, 2006. Disponível em: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>. Acesso em: 15 jan. 2007.
- [SIL07] SILVEIRA, P. **Processo de ETC orientado a serviço para um ambiente de gestão de qualidade de software**. 2007. 168 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2007.
- [SOF07] SOFTEX. **MPS.Br Capacitação e Empreendedorismo**. Disponível em: http://www.softex.br/mpsbr/_home/default.asp. Acesso em: 20 nov. 2007.
- [SOM04] SOMMERVILLE, I. **Software Engineering**. 5th Edition. Boston: Addison-Wesley, 2004. 592 p.
- [SQL07] **SQL Server 2000 – Data Transformation Services**. Disponível em: <http://technet.microsoft.com/en-us/sqlserver/bb331744.aspx>. Acesso em: 25 nov. 2007.

- [SUB99] SUBRAMANYAM, V.; SHARMA, S. **HPD - Query tool on Projects Historical Database**. Hewlett-Packard – Latin American Software Operation (LASO). Porto Alegre, Hewlett Packard, 1999.
- [TAN06] TAN, P.N.; Steinbach, M.; KUMAR, V. **Introduction to Data Mining**. Boston: Addison Wesley, 2006. 769 p.
- [WIN07] WINCK, A.T. **Um Processo de KDD para auxílio à reconfiguração de ambientes virtualizados**. 2007. 78 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Informática, PUCRS, Porto Alegre, 2007.
- [WIT05] WITTEN, I.; FRANK, E. **Data mining: practical machine learning tools and techniques**. San Francisco: Morgan Kaufmann, 2005. 525 p.

APÊNDICE A – Arquivo *Arff*

A { @relation Esforço

B { @attribute Tamanho numeric
 @attribute Num_Usuarios {1,2,3,4,5,Mais_5}
 @attribute Causa_Raiz {Falta_atenção_envolvido,Problemas_Colaborador,
 Especificação_documentação,
 Procedimentos_Qualidade,Outros,Problema_Não_Identificado,
 Falha_Lógica}
 @attribute Fase_Origem {Analise_CEF,Client,Projeto,Server,Teste}
 @attribute TipoBase {DEF_PRE_REL,DI,DEF_POS_REL}
 @attribute Severidade numeric

C { @attribute Classe_Esforco {1_5Horas,]1_5_a_4Horas,]4_a_6Horas,]Mais_6Horas}

D { @data
 31.8,1,?,?,DI,2,1_5Horas
 117.66,1,?,?,DI,2,1_5Horas
 12.72,1,Procedimentos_Qualidade,Server,DI,2,1_5Horas
 30.74,1,Falta_atenção_envolvido,Projeto,DI,1,1_5Horas
 64.66,1,Procedimentos_Qualidade,Teste,DI,1,1_5Horas
 69.96,1,Especificação_documentação,Client,DI,2,1_5Horas
 76.32,1,Especificação_documentação,Teste,DI,2,1_5Horas
 84.8,1,Procedimentos_Qualidade,Projeto,DI,2,1_5Horas
 101.76,1,Outros,Client,DI,2,1_5Horas
 101.76,1,?,Teste,DI,2,1_5Horas
 102.82,1,Procedimentos_Qualidade,Teste,DI,1,1_5Horas
 5.3,1,Falta_atenção_envolvido,Client,DI,3,1_5Horas
 6.36,1,Procedimentos_Qualidade,Teste,DI,2,1_5Horas
 6.36,1,Problemas_Colaborador,Client,DI,2,1_5Horas
 6.36,1,Especificação_documentação,Teste,DI,1,1_5Horas
 13.78,1,Falta_atenção_envolvido,Client,DEF_PRE_REL,3,1_5Horas
 13.78,1,Falta_atenção_envolvido,Client,DEF_PRE_REL,3,1_5Horas
 20.14,1,Falta_atenção_envolvido,Projeto,DI,4,1_5Horas
 30.28,1,Falta_atenção_envolvido,Server,DI,3,1_5Horas
 30.74,1,Procedimentos_Qualidade,Projeto,DI,2,1_5Horas
 33.92,2,Falta_atenção_envolvido,Teste,DI,3,1_5Horas
 44.52,1,Procedimentos_Qualidade,Teste,DI,1,1_5Horas
 48.76,2,Outros,Server,DEF_PRE_REL,3,1_5Horas
 3.18,3,Falha_Lógica,Client,DEF_PRE_REL,3,Mais_6Horas
 4.24,Mais_5,Especificação_documentação,Analise_CEF,DEF_PRE_REL,4,Mais_6Horas
 6.36,3,?,?,DI,3,Mais_6Horas
 6.36,4,?,?,DEF_PRE_REL,3,Mais_6Horas
 6.36,4,Procedimentos_Qualidade,Client,DEF_PRE_REL,4,Mais_6Horas
 18.02,3,?,?,DI,4,Mais_6Horas
 18.02,4,?,?,DEF_PRE_REL,4,Mais_6Horas
 30.74,4,Falta_atenção_envolvido,Client,DI,4,Mais_6Horas
 31.8,1,?,?,DI,1,Mais_6Horas
 34.98,4,Procedimentos_Qualidade,Server,DEF_PRE_REL,3,Mais_6Horas

A – Nome do arquivo *ARFF*

B – Atributos Explanatórios

C – Atributo Classe

D – Instâncias (Registros) a serem Mineradas