FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
CENTRO UNIVERSITÁRIO "EURÍPIDES DE MARÍLIA" – UNIVEM
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDERSON MARCIANO
MERLEY DA SILVA CONRADO

FERRAMENTA DE APOIO E AVALIAÇÃO EM TESTE DE SOFTWARE UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS

MARÍLIA
2006

MERLEY DA SILVA CONRADO
ANDERSON MARCIANO

FERRAMENTA DE APOIO E AVALIAÇÃO EM TESTE DE SOFTWARE UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS

Monografia apresentada ao Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha, para obtenção do Título de Bacharel em Ciência da Computação.

Orientador:
Prof. Dr. Edmundo Sérgio Spoto

MARÍLIA
2006

MARCIANO, Anderson e CONRADO, Merley da Silva

Ferramenta de apoio e avaliação em teste de software utilizando técnicas de mineração de dados / Anderson Marciano e Merley da Silva Conrado; orientador: Edmundo Sérgio Spoto. Marília, SP: dezembro, 2006.

77 f.

Monografia (Bacharel em Ciência da Computação) – Centro Universitário Eurípides de Marília - Fundação de Ensino Eurípides Soares da Rocha.

1.Mineração de dados 2.Teoria da Utilidade
3.Teoria da Probabilidade.

CDD: 005.1

ANDERSON MARCIANO

FERRAMENTA DE APOIO E AVALIAÇÃO EM TESTE DE SOFTWARE UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS

Banca examinadora da monografia apresentada ao Curso de Ciência de Computação da UNIVEM,/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação. Área de Concentração: Engenharia de Software.

Resultado: _____

ORIENTADOR: Prof. Dr. Edmundo Sérgio Spoto

1º EXAMINADOR: Kalinka Regina L. Jaquie Castelo Branco

2º EXAMINADOR: Márcio Eduardo Delamaro

Marília, 05 de dezembro de 2006.

MERLEY DA SILVA CONRADO

FERRAMENTA DE APOIO E AVALIAÇÃO EM TESTE DE SOFTWARE UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS

Banca examinadora da monografia apresentada ao Curso de Ciência de Computação da UNIVEM,/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação. Área de Concentração: Engenharia de Software.

Resultado: _____

ORIENTADOR: Prof. Dr. Edmundo Sérgio Spoto

1º EXAMINADOR: Kalinka Regina L. Jaquie Castelo Branco

2º EXAMINADOR: Márcio Eduardo Delamaro

Marília, 05 de dezembro de 2006.

Dedicamos este trabalho aos nossos pais pelos incentivos incansáveis e paciência em tolerar a nossa ausência.

AGRADECIMENTOS

Primeiramente agradecemos muito a Deus, silencioso e fundamental esteio de tudo!

Ao nosso orientador, Prof. Dr. Edmundo Sérgio Spoto, por enriquecer este trabalho com seu vasto conhecimento e presteza no desenvolvimento e finalização do mesmo.

Agradecemos ao corpo docente e a todos que contribuíram nesta etapa da nossa vida.

Agradeço aos meus queridos pais, Sidnei e Sueli.

A.M.

Agradeço aos meus amados pais, Gecé e Lindinalva, aos meus irmãos, à minha irmã Midiam, ao meu cunhado Caio, ao doce brilho dos olhos da minha sobrinha Tássyla e ao Anderson.

À Profª. Ana Paula Perruza, pela sua atenção e simpatia. A todos meus amigos, Aline, Andréia, Dani, Gaby, Laura... e familiares.

M.S.C.

MARCIANO, Anderson; CONRADO, Merley da Silva**. Ferramenta de apoio e avaliação em teste de software utilizando técnicas de mineração de dados**. 2006. 77 f. Trabalho de conclusão de curso. Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, 2006.

RESUMO

Neste trabalho são relatadas as técnicas de mineração de dados (*data mining*) visando aprimorar as escolhas de casos de testes gerados automaticamente a partir de uma ferramenta de geração de casos de testes utilizando técnicas baseado na especificação. São apresentados os conceitos básicos de técnicas de mineração e dados como a teoria da probabilidade e a teoria da utilidade, sendo estas utilizadas na confecção de uma ferramenta que auxilia na escolha dos casos de testes que serão reaplicados nas avaliações dos resultados dos testes em programas espaciais. Este trabalho apresenta resultados de uso das técnicas utilizadas.

Palavras-chave: Mineração de dados. Teoria da Utilidade. Teoria da Probabilidade.

MARCIANO, Anderson; CONRADO, Merley da Silva**. Ferramenta de apoio e avaliação em teste de software utilizando técnicas de mineração de dados**. 2006. 77 f. Trabalho de conclusão de curso. Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, 2006.

ABSTRACT

In this work the techniques of data mining are told aiming at to improve the choices of tests cases generated automatically from a tool of generation of tests cases being used techniques based in the specification. The basic concepts of data mining techniques are presented as the probability theory and the utility theory, being these used in the confection of a tool that it assists in the choice of the tests cases that will be reapplied in the evaluations of the results of the tests in space programs. This work presents resulted of use of the used techniques.

Keywords: Data mining. Utility Theory. Probability Theory.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

INTRODUÇÃO

A área de Engenharia de *Software* busca melhorar as técnicas de validação, verificação e teste de *software* para elevar a qualidade do *software*. Várias técnicas de teste de s*oftware* são propostas e avaliadas pela literatura e a grande dificuldade encontrada geralmente é a falta de um mecanismo automático de escolher e comparar qual técnica melhor se enquadra ao tipo de teste que está sendo proposto e quais resultados tais técnicas proporcionam nos devidos testes.

Teste de *software* deve ser apoiado por ferramentas que auxiliam na geração dos casos de testes, bem como na busca de avaliações por casos de testes que tenham um melhor resultado. Visando centralizar as diferentes técnicas de teste, bem como armazenar os diferentes resultados obtidos, foi proposta, neste trabalho, a criação de uma ferramenta que armazene todos os casos de testes gerados nas etapas de testes (em cada técnica utilizada) bem como os resultados obtidos em cada execução do teste. O teste de um programa, segundo Pressman (1991), é o ato de executar um programa com o intuito de detectar erros.

A Ferramenta de teste do PLAVIS (*Plataform for software Validation and Integration for Space systems*) gera automaticamente conjuntos muito grandes de casos de teste, utilizando outras ferramentas que geram casos de testes de maneira automática. Para que a escolha de características desejadas de casos de testes, é utilizada a técnica de mineração de dados aplicada no conjunto de casos de teste gerados pela Ferramenta de teste do PLAVIS.

Objetivos

Com a existência da ferramenta que centraliza as diferentes técnicas de teste e armazena os resultados (sendo que muitos são gerados automaticamente), torna-se difícil para o usuário buscar uma avaliação com mais rigor e comparar qual tipo de técnica gera melhores resultados. Uma das maneiras encontradas para melhorar as avaliações e resultados é criar uma ferramenta, que tem como intuito minerar todas as informações necessárias de teste utilizando assim técnicas especiais de mineração de dados.

Para que isso seja possível, é criado um *data warehouse* (armazém de dados), conforme descrito na Seção 1.2, de informações de testes que armazenará todos os casos de testes e os resultados obtidos em cada execução utilizando a técnica de mineração de dados, para, então, apoiar a avaliação de teste de *software*.

Na construção da ferramenta de mineração de dados são implementadas as funcionalidades da teoria da utilidade, que contribui com a representação e raciocínio da decisão da preferência do testador. A partir das técnicas da teoria da probabilidade e da utilidade foi possível aplicar funcionalidades de decisões geradas pelas necessidades do usuário de teste.

Organização do Trabalho

Este trabalho está organizado da seguinte maneira:

No Capítulo 1 são apresentados os conceitos e terminologias que adotados neste trabalho, sendo composto pela Seção 1.1 que descreve a mineração de dados, ressaltando também suas principais tarefas.

Na Seção 1.2 é definido brevemente *data warehouse* e na Seção 1.3 é apresentado o conceito da teoria da decisão, citando os principais métodos para a

escolha da melhor decisão. Na Seção 1.4, desenvolvida por Merley da Silva Conrado, é descrita a Teoria da Utilidade, como seus conceitos e funcionamento.

Por fim, na Seção 1.5, desenvolvida por Anderson Marciano, são apresentados os conceitos da Teoria da Probabilidade e também seu funcionamento.

No Capítulo 2 são apresentadas as tecnologias utilizadas, bem como o motivo de tal escolha.

No Capítulo 3 é mostrado como a ferramenta proposta neste trabalho foi desenvolvida. Na Seção 3.1 são expostos os objetivos e a arquitetura da ferramenta, na Seção 3.2 são apresentadas as funcionalidades da ferramenta, bem como a interface e a interação com o usuário, na Seção 3.3 é explicado como foi feita a extração de dados da Ferramenta PLAVIS e na Seção 3.4 como foi feito o armazenamento dos dados. Na Seção 3.5 são apresentados os cálculos probabilísticos, na Seção 3.6, desenvolvida por Anderson Marciano, são apresentadas as opções de métodos disponíveis para o decisor, na Seção 3.7, desenvolvida por Merley da Silva Conrado, é apresentada como é feita a busca de melhor resultado e, por fim, na Seção 3.8 são expostas as considerações finais deste capítulo.

No Capítulo 4 são apresentados os resultados obtidos.

No último capítulo são expostas as conclusões baseadas no desenvolvimento deste trabalho, como também os trabalhos futuros.

1 CONCEITOS E TERMINOLOGIAS

1.1 Mineração de dados

1.1.1 Conceitos

O aumento da necessidade de se desenvolver novas ferramentas e técnicas de extração de conhecimento a partir de dados armazenados é devido ao elevado volume de dados, possivelmente úteis, armazenados nas bases de dados em geral (REZENDE et al, 2003).

Com a queda dos preços de computadores, o armazenamento de dados se tornou um método eficaz e muito praticado nos dias de hoje, tornando viável o custo do armazenamento em base de dados. Devido a isto, hoje é fácil encontrar elevadas quantidades de dados espalhadas em vários setores como bolsa de valores, mercados, produção, médicos, comércio lojista e científico.

Para descobrir informações ocultas nestes dados, é utilizado um conceito criado no final da década de oitenta, denominado como mineração de dados (REZENDE et al, 2003). Esta explora e analisa, por meios automáticos ou semi-automáticos, repositórios para descobrir padrões e regras significativas (BERRY e LINOFF, 1997), como, por exemplo, padrões de pacientes que desenvolveram doenças (que podem ser úteis na tentativa de preverem diagnósticos e antecipar tratamentos), perfis de compra de clientes (para usar em futuras grandes promoções).

Zhou (2003) avalia três diferentes perspectivas desta área, Banco de Dados, Aprendizado de Máquina e Estatística. Sendo que neste trabalho é utilizada a

perspectiva do Banco de Dados, que ressalta a eficiência no processo da descoberta do conhecimento em uma enorme quantidade de dados.

Rezende et al (2003) descrevem que na caracterização da área de mineração de dados é importante, também, discutir a caracterização do que na literatura é chamado de KDD (*Knowledge Discovery in Databases*). De acordo com W. Frawley (1992), KDD é a "extração de conhecimento previamente desconhecido, implícito e potencialmente útil, a partir de dados". E segundo Fayyad, Piatetsky-Shapiro, e Smyth (1996) KDD é definido como sendo um processo não trivial de identificação de padrões válidos, modernos e potencialmente úteis.

> "O foco central de mineração de dados é o de como transformar dados armazenados em conhecimento, expresso em termos de formalismos de representação, tal como regras e relações entre dados. Existe conhecimento que pode ser extraído diretamente de dados sem o uso de qualquer técnica, entretanto, existe também muito conhecimento que está de certa forma "embutido" na base de dados, na forma de relações existentes entre itens de dados que, para ser extraído, é necessário o desenvolvimento de técnicas especiais" (REZENDE et al 2003).

Feelders, Daniels e Holsheimer afirmaram que recentemente tem sido argumentado que a escolha do algoritmo de mineração de dados pode não ser adequada para a obtenção da informação desejada, quando usado para fazer suporte à decisão do mundo real. As limitações estão diretamente ligadas a uma variedade de fatores tais como:

- a incompatibilidade do domínio do conhecimento com modelo de transação da base de dados, tais como conhecimento embutido em corporação com regras políticas e regulamento de negócios;
- dificuldade na representação como dúvida, imprecisão ou conhecimento incompleto sobre o fenômeno considerado;
- em algumas aplicações, gasto computacional;

Na Seção 1.1.2 são definidos os conceitos básicos de dados, informação e conhecimento que são considerações importantes para o processo de desenvolvimento de mineração de dados.

1.1.2 Dados, Informação e Conhecimento

Os conceitos de dados, informação e conhecimento estão interligados. Na Figura 1.1 é mostrada uma representação gráfica do relacionamento entre dados, informação e conhecimento, em função da capacidade de entendimento e da independência de contexto que cada um destes conceitos implica (REZENDE et al, 2003).

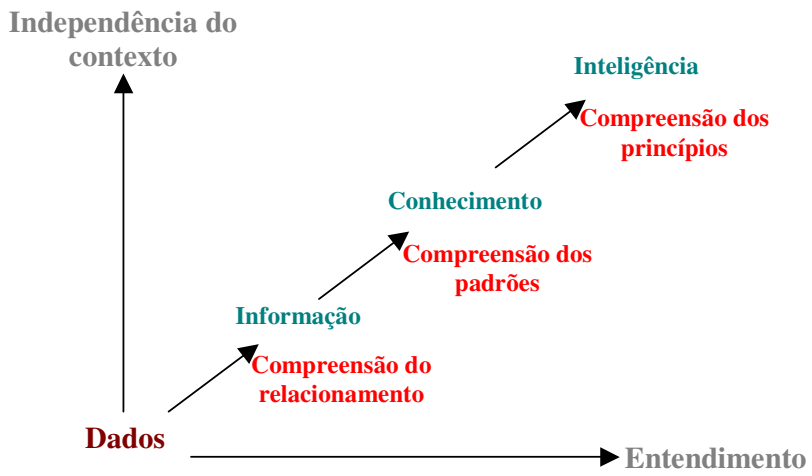


Figura 1.1 - Dado, informação e conhecimento (REZENDE et al, 2003).

Em seguida será explicado o que significa cada um deles: dado, informação e conhecimento (REZENDE et al, 2003).

O *dado* é um elemento puro, em seu estado primário. Dados são fatos, números, texto ou qualquer mídia que possa ser processada pelo computador.

Entre os dados armazenados atualmente estão:

• Dados operacionais ou transacionais como vendas, custos, inventários, folhas de pagamento ou contas correntes;

• Dados não operacionais como previsões de mercado, vendas ao nível industrial, e dados macro-econômicos;

• Metadados, ou dados descrevendo a estrutura dos dados armazenados, tais como projetos lógicos de bancos de dados ou dicionários de dados;

• Mídia contendo imagens, sons ou uma combinatória de ambos, que além de ser armazenada, pode ser catalogada eletronicamente.

A *informação* é um conjunto de dados no qual o ser humano atribui um significado. Ela pode gerar conhecimento que ajude na análise de padrões históricos para conseguir uma previsão dos fatos futuros. Por exemplo, a informação dos dados sumarizados nas vendas de um determinado ambiente comercial pode ser analisada com a finalidade de fornecer informações relacionadas com a natureza dos clientes.

O *conhecimento* é a capacidade de resolver problemas, inovar e aprender baseando-se em experiências prévias. O conhecimento refere-se à habilidade de criar um modelo mental que descreva o objeto e indique as ações a programar e as decisões a tomar. O conhecimento pode ser representado como uma combinação de estruturas de dados e procedimentos interpretáveis que levam a um comportamento conhecido.

Este comportamento fornece informações que podem, então, ser utilizadas para planejar e decidir.

A compreensão, análise e síntese, necessárias para a tomada de decisões inteligentes, são realizadas a partir do nível do conhecimento. Assim, é fundamental que se mantenha a coerência dos dados que estão armazenados nos diferentes repositórios e

das informações nos diferentes níveis (REZENDE et al, 2003). Algumas das áreas que trabalham com dados, informação e conhecimento são: *marketing*, investimento, detecção de fraude, manufatura, telecomunicações e muitos outros.

1.1.3 Generalidades de Mineração de Dados

Para uma melhor compreensão da definição de mineração de dados que foi citado nas seções anteriores, deve-se entender cada item abaixo (REZENDE et al, 2003):

Dados - Conjunto de fatos ou casos em um repositório de dados;

Padrões - Denotam alguma abstração de um subconjunto dos dados em alguma linguagem descritiva de conceitos;

Processo - A Extração de Conhecimento de Base de Dados envolve diversas etapas como a preparação dos dados, busca por padrões e avaliação do conhecimento;

Válidos - Os padrões descobertos devem possuir algum grau de certeza, ou seja, devem satisfazer funções ou limiares que garantam que os exemplos cobertos e os casos relacionados ao padrão encontrado sejam aceitáveis;

Novos - Um padrão encontrado deve fornecer novas informações sobre os dados. O grau de novidade serve para determinar quão novo ou inédito é um padrão. Pode ser medido por meio de comparações entre as mudanças ocorridas nos dados ou no conhecimento anterior;

Úteis - Os padrões descobertos devem ser incorporados para serem utilizados;

Compreensíveis - Um dos objetivos de realizar MD é encontrar padrões descritos em alguma linguagem que pode ser compreendida pelos usuários permitindo uma análise mais profunda dos dados;

Conhecimento - O conhecimento é definido em termos dependentes do domínio, relacionados fortemente com medidas de utilidade, originalidade e compreensão.

O processo de Extração de Conhecimento de Bases de Dados tem por finalidade descobrir o conhecimento que está escondido na base de dados para que possa ser utilizado em um caso de decisão, portanto é necessário que as informações sejam extraídas de forma correta para que o usuário final tenha confiança nas informações e que consista nas informações que deseja.

1.1.3.1 Processo de Mineração de Dados

Para o desenvolvimento deste trabalho foram consideradas três etapas de processo: pré-processamento, extração de padrões e pós-processamento. Na Figura 1.2 são ilustradas tais etapas.

Portanto, o processo de mineração de dados começa com o entendimento da aplicação, considerando os aspectos como objetivos da aplicação e as fontes de dados, da qual se pretende extrair o conhecimento. Em seguida, é realizada uma seleção dos dados a partir dessas fontes, de acordo com os objetivos do processo. Os conjuntos de dados que estão nesta seleção são, então, pré-processados, recebem um tratamento para poderem ser submetidos aos métodos e ferramentas na etapa de extração de padrões. A etapa de extração de padrões tem o objetivo de encontrar modelos (conhecimento) a partir de dados. Após a etapa de extração de padrões, vem a de pós-processamento, na qual o conhecimento é avaliado quanto a sua qualidade e/ou utilidade para que, em caso positivo, seja utilizado para apoio a algum processo de tomada de decisão (REZENDE et al, 2003).

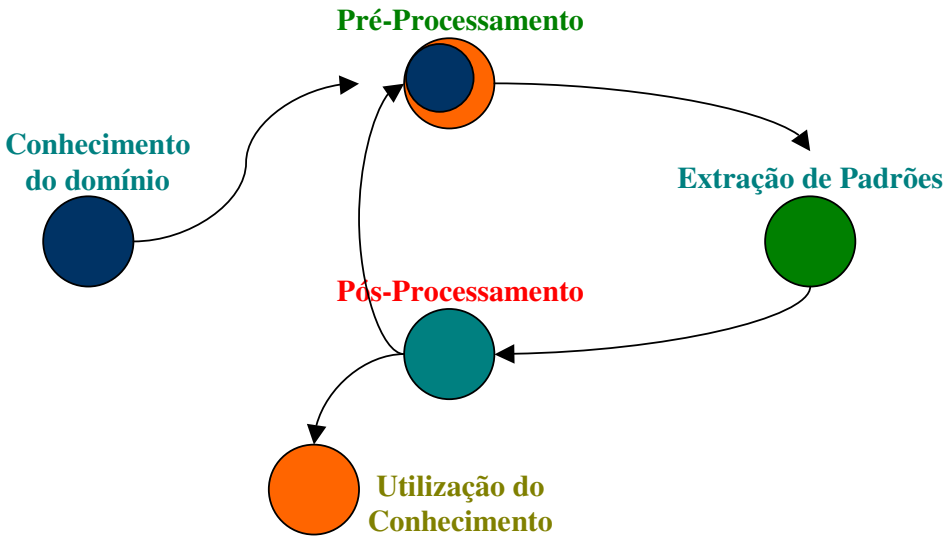


Figura 1.2 - Etapas do processo de Mineração de dados (REZENDE et al, 2003).

- **Pré-Processamento**

Normalmente os dados que estão disponíveis para análise não estão em um formato adequado para a extração de conhecimento. Além disso, possui-se uma quantidade mínima de memória ou tempo de processamento, para que possa fazer uma aplicação direta dos algoritmos de extração de padrões nos dados. Desse modo, necessita-se da aplicação de métodos para o tratamento, limpeza e redução do volume de dados antes de começar a etapa de extração de padrões. É importante que os dados que serão transformados tenham os objetivos cumpridos.

Existem muitas transformações de dados que podem ser executadas na etapa de pré-processamento dos dados, entre elas: extração e integração, transformação, limpeza, seleção e redução de dados (REZENDE et al, 2003). Porém, a transformação de dados utilizada no processo de desenvolvimento da ferramenta deste trabalho foi a seleção,

que teve como finalidade permitir que o usuário selecione os dados de acordo com a preferência para serem extraídos e unificados em uma única tabela, que servirá como entrada no algoritmo de extração de padrões (Ver APÊNDICE A – Intersecções dos dados).

- **Extração de Padrões**

A extração de padrões tem por finalidade o cumprimento dos objetivos definidos na identificação do problema para que possa colher a informação desejada. Nesta etapa tem que escolher qual o algoritmo vai ser usado para extrair as informações que se deseja obter.

- **Escolha da tarefa**

A escolha da tarefa é realizada conforme o objetivo a ser alcançado. As tarefas possíveis de um algoritmo de extração de padrões podem ser agrupadas em atividades preditivas e descritivas. Atividades de predição, ou mineração de dados preditivo, consistem na generalização de exemplos ou experiências passadas com respostas conhecidas em uma linguagem capaz de reconhecer a classe de um novo exemplo. Os dois principais tipos de tarefas para predição são classificação e regressão. A classificação consiste na predição de um valor categórico como, por exemplo, predizer se o cliente é bom ou mau pagador. Na regressão, o atributo a ser predito consiste em um valor contínuo como, por exemplo, predizer o lucro ou a perda em um empréstimo (WEISS e INDURKHYA, 1998) *apud* (REZENDE et al, 2003).

Atividades de descrição, ou Mineração de dados descritivos, consistem na identificação de comportamentos intrínsecos do conjunto de dados, sendo que estes dados não possuem uma classe especificada. Algumas das tarefas de descrição são *clustering*, regras de associação e sumarização.

Uma vez eleita a tarefa a ser empregada, existe uma variedade de algoritmos para executá-la. A definição do algoritmo de extração e a posterior configuração de seus parâmetros também são realizadas nesta etapa.

- **Escolha do algoritmo**

A escolha do algoritmo é realizada de forma subordinada à linguagem de representação dos padrões a serem encontrados. Entre os tipos mais freqüentes de representação de padrões, destacam-se:

- Árvores de decisão;
- Regras de produção;
- Modelos lineares;
- Modelos não-lineares (redes neurais artificiais);
- Modelos baseados em exemplos (knn — *k-nearest neighbor*, raciocínio baseado em casos);
- E modelos de dependência probabilística (Redes Bayesianas).

Kearns e Vazirani (1994 *apud*, REZENDE, 2003), apresentam uma sugestão para escolher uma determinada função: o modelo mais apropriado é aquele mais simples que seja consistente com todas as observações. Normalmente, soluções mais complexas são preferidas por pesquisadores, enquanto os práticos tendem a preferir modelos mais

simples em virtude de sua fácil interpretação (FAYYAD, PIATETSKY-SHAPIRO, & SMYTH, 1996).

Portanto, neste trabalho, foi utilizado o modelo de dependência probabilística que são as Redes Bayesianas, cujo funcionamento desta técnica é explicado com detalhes na Seção 1.5.3.

- **Pós-Processamento**

O conhecimento obtido pode ser utilizado para solucionar diversos problemas que ocorre no decorrer da vida, seja ele resolvido por um Sistema Inteligente ou por um ser humano no apoio a algum processo de tomada de decisão. Para isso é importante que algumas questões sejam respondidas aos usuários (LIU e HSU, 1996):

- O conhecimento extraído representa o conhecimento do especialista?
- De que maneira o conhecimento do especialista difere do conhecimento extraído?
- Em que parte o conhecimento do especialista está correto?

No entanto, é de vital importância desenvolver algumas técnicas de apoio no sentido de fornecer aos usuários apenas os padrões mais interessantes (SILBERSCHATZ e TUZHILIN, 1995).

Entretanto, podem ocorrer casos em que os modelos são muito complexos ou não fazem sentido para os especialistas (PAZZANI, MANI e SHANKLE, 1997).

Portanto os dados têm que ser de fácil interpretação e que revele importância para os seres humanos.

1.1.3.2 Principais Tarefas de Mineração de Dados

A seguir são citadas algumas técnicas que são utilizadas para extrair diferentes tipos de conhecimento, pois nos dias de hoje com a grande diversidade de dados torna-se importante o conhecimento delas para que o desenvolvedor de *software* possa aplicar a técnica de forma a obter o conhecimento que deseja.

- **Classificação**

Classificação é o processo de encontrar um modelo que descreva classes diferentes de dados. As classes são predeterminadas (ELMASRI e NAVATHE, 2005).

Por exemplo, em uma aplicação em que a loja deseja saber se um cliente é confiável, ela então o classifica em 'Excelente Cliente', 'Bom Cliente' e 'Mau Cliente', evitando que a loja obtenha perdas com estes clientes.

- **Regressão**

Regressão é uma aplicação especial da regra de classificação. Se uma regra de classificação é considerada uma função sobre variáveis que as mapeia em uma classe destino, a regra é chamada regressão (ELMASRI e NAVATHE, 2005).

O objetivo da tarefa regressão é encontrar uma relação entre um conjunto de atributos de entrada (variáveis de entrada ou variáveis preditoras) e um atributo-meta contínuo. Sejam X = {x1,..., xd} os atributos de entrada e y o atributo-meta, o objetivo é encontrar um mapeamento da seguinte forma:

$$y = f(x_1, x_2, \ldots, x_d)$$

Por exemplo, se a faixa de idade está entre 1 e 100 e o número de crianças entre 1 e 5, pode-se medir estes campos e constatar que eles podem fazer parte dentro de um mesmo intervalo, por exemplo de -1 a +1.

- **Regras de Associação**

Associação é uma das principais tecnologias de mineração de dados. Uma regra de associação caracteriza o quanto a presença de um conjunto de itens nos registros de uma Base de Dados implica na presença de algum outro conjunto distinto de itens nos mesmos registros (AGRAWAL e SRIKANT, 1994).

Por exemplo, observando os dados de vendas de uma loja de eletroeletrônica sabe-se que 86% dos clientes que compram uma televisão também adquirem, na mesma compra, um vídeo DVD. Nessa regra, 86% correspondente a sua confiabilidade.

- **Agrupamento**

O objetivo da técnica de agrupamento é colocar os registros em grupos, de tal forma que os registros de um grupo sejam similares aos demais do mesmo grupo e diferentes dos demais grupos (ELMASRI e NAVATHE, 2005).

Por exemplo, em uma loja pode-se agrupar as pessoas que possuem um padrão financeiro elevado, isolando-as das que não possuem, para que a loja possa enviar uma promoção de produtos somente para as pessoas de padrão financeiro elevado.

1.2 *Data warehouse* (DW)

1.2.1 Conceito de DW

Não há uma definição única do termo *data warehouse* (DW), pelo fato do mesmo ser desenvolvido por muitas organizações para satisfazer necessidades específicas. Portanto, ele foi definido como uma coleção de dados integrada, isto é, que se refere a toda organização, disponíveis para consultas, análises e orientadas ao negócio, o que significa que DW é modelado em função das necessidades. Pode ser usado para bases de dados muito grandes, distribuídas, heterogêneas e outras fontes de informação.

1.2.2 Aplicabilidade do DW

Pode-se considerar que o *data warehouse* (DW) provê novas maneiras de interação, manipulação, controle e análise completa dos dados. Proporciona também descoberta de conhecimento, possibilitando que os mesmos se transformem em informações que auxiliem o processo decisório, especialmente para preparação dos dados para mineração de dados.

O DW distingue-se dos outros bancos de dados por direcionar principalmente para aplicações de apoio às decisões, sendo este um dos motivos de seu uso neste trabalho.

A informação selecionada é extraída, filtrada conforme a necessidade e, assim, armazenada em um repositório, de forma a facilitar a visualização e o acesso dos dados.

Quando uma consulta é feita, esta é avaliada diretamente no repositório (DW) sem alcançar as fontes originais de informação..

1.3 Teoria da Decisão

1.3.1 Origem

Nos anos 70 surgiram os primeiros métodos voltados para os problemas de decisão discretos, que segundo Gomes e Freitas Jr. (2000, p. 85), são aqueles que tratam de um número finito de alternativas, existindo também os contínuos (quando tal número pode ser pensado como sendo infinitamente grande).

Por meio de pesquisa em métodos sistemáticos surgiu a Teoria de Decisão, que tem como objetivo escolher entre vários caminhos existentes, usando um volume muito elevado de informações disponíveis, aquele que mais se adequar, isto é, a "melhor decisão lógica possível" de acordo com as preferências do decisor, tendo como resultado a perfeição (resultado certo) e a imperfeição (resultado incerto). Porém, não se garante que a aplicação desta teoria produzirá resultados ótimos, mas diminui a chance de obter um resultado desfavorável.

Este contexto é baseado na teoria de probabilidades (para atribuir crenças a informações incompletas) e na teoria de utilidade (para manter consistência entre as preferências do decisor e as decisões tomadas). Assim, tomar decisões significa escolher as decisões que maximizam a utilidade esperada dos resultados em um conjunto de decisões com resultados incertos.

1.3.2 Métodos para a Escolha da Melhor Decisão

A melhor decisão (estratégia) é sempre aquela que garanta o melhor resultado possível, devendo ter a maior quantidade de informações possíveis referente a cada alternativa.

Existem vários métodos de auxílio de tomada de decisões. Alguns destes métodos apresentam modelos matemáticos complicados e dependem da determinação de parâmetros subjetivos ou da realização de rotinas matemáticas complicadas. Sendo estes alguns dos motivos que levam às empresas a não utilizarem freqüentemente essas metodologias e a continuarem usando métodos tradicionais de decisão, que dependem, em sua maioria, do *feeling* do tomador de decisão (GUGLIELMETTI, MARINS E SALOMON, 2003).

Com o desenvolvimento da computação, obteve-se uma melhora na interface do tomador de decisão, tornando-a mais amigável, permitindo que o mesmo expresse com clareza as suas preferências, sem a necessidade de pensar no algoritmo matemático do método utilizado (PINHO et al. 1996) *apud* (GUGLIELMETTI, MARINS e SALOMON, 2003).

Dentre os métodos existentes para a escolha da melhor decisão, podem-se citar:

- ***Promethee***

Este método foi desenvolvido por Brans e Vincke, em 1985. É adequado às situações em que os critérios possam ser representados em forma de valores.

Para dada alternativa "A", sua ordenação é obtida pela classificação de seu somatório líquido, ou seja, o somatório das preferências de "A" sobre todas as demais alternativas, menos o somatório das preferências das demais alternativas sobre "A".

Como vantagens deste método podem-se citar:

- considera as regras de dominância;
- trabalha com matemática simples, com lógica compreensível para muitos tomadores de decisão;
- devido às funções de preferência, permite opções de relacionamento entre as alternativas. Porém há desvantagens, como:
- necessidade de transformar critérios qualitativos em valores, tendo em vista que isto pode ser possibilitado pela elevada variedade de funções, mas com um grau de habilidade;
- na análise de sensibilidade, a mudança de pontuação final derivada da alteração de uma hipótese pode não ser adequadamente percebida pelo decisor.

- ***Elimination et Choix Traduisant la Realité*** **(ELECTRE)**

É um método de auxílio à tomada de decisão por múltiplos critérios. Devido a sua classificação conforme o critério da dominância de uma alternativa sobre outra, este método reduz o tamanho do conjunto de possíveis alternativas. Um método semelhante é o *Promethee*.

Uma desvantagem é a necessidade de transformar os valores cardinais em ordinais.

- ***Analytic Hierarchy Process* (AHP)**

Criado na década de 70 por Saaty, o método de auxílio à tomada de decisão por múltiplos critérios, AHP, tem início na estruturação do problema que começa na definição de um objetivo global, que, em seguida, é definido os sub-objetivos numa estrutura de árvore, tendo como raiz o objetivo global. Conforme se afastam da raiz os fatores se tornam mais específicos. As extremidades, as folhas da árvore, representam os critérios ou objetivos. O ANP é uma extensão deste método.

O número feito de comparações necessárias para a elucidação das preferências é uma das limitações apresentadas pelo AHP, sendo só permitido fazer comparação par-a-par, levando em alguns casos a um número muito elevado de julgamentos.

Outra desvantagem deste método é que o mesmo tem restrição na quantidade de alternativas e critérios.

Porém, de acordo com a pesquisa feita em Pontifícia Universidade Católica do Paraná, em Curitiba em 2002, em que os métodos AHP, MAHP e ELECTE I foram comparados pelos seus usuários, o AHP foi considerado o mais amigável dentre os três, principalmente no que diz respeito à execução de aplicações práticas (GUGLIELMETTI, MARINS E SALOMON, 2003), mas não foi encontrada nesta pesquisa uma comparação com a Teoria da Utilidade.

- **MAHP**

Um método de auxílio à tomada de decisão por múltiplos critérios que tem como desvantagem a necessidade de fazer a transformação para a escala geométrica.

Abaixo é apresentado um quadro comparando os métodos AHP, MAHP E ELECTRE I, quanto a interface tomador de decisão, de Guglielmetti, Marins e Salomon (2003).

Tabela 1 - Comparação teórica entre a Interface tomador de decisão *versus* método

	AHP	**MAHP**	**ELECTRE I**
Interface tomador de decisão *versus* método			
Disponibilidade de *software* para *download* gratuito	**Sim**	**Não**	**Não**
Necessidade de um especialista no método utilizado	Média	Alta	Média
Utilização de decisões em grupo	**Sim**	**Sim**	**Não**
Permissão para a participação de mais de uma pessoa na decisão	**Sim**	**Sim**	**Sim**
Facilidade para estruturar o problema	**Alta**	**Média**	**N/A**
Possibilita o aprendizado sobre a estrutura do problema	**Sim**	**Sim**	**N/A**
Nível de compreensão conceitual e detalhada do modelo e algoritmo	**Alto**	**Médio**	**Baixo**
Nível de compreensão para o decisor referente à forma de trabalho	**Alto**	**Médio**	**Baixo**
Transparência no processamento e nos resultados	**Alta**	**Baixa**	**Média**
Quantidade de aplicações práticas	**Alta**	**Baixa**	**Baixa**
Número de publicações científicas	**Alta**	**Baixa**	**Média**

- ***Measuring Attractiveness by a Categorical Based Evaluation Technique* (MACBETH)**

Método que critica o uso de funções com valor cardinal. Fornece um indicador de inconsistência do conjunto de critérios formulados, facilitando sua revisão por Programação Linear.

A mesma limitação citada no método AHP aparece neste método, porém em maior grau, pois quanto maior o número de critérios a serem avaliados, são necessárias mais comparações.

- **Teoria da Utilidade**

Segundo Russell e Norvig (2004), este método descreve o que um decisor deve considerar para se tomar uma decisão com base na evidência. Permite o tratamento de múltiplos critérios e a inserção de variáveis qualitativas, assim como o método AHP.

Devido às desvantagens de outros métodos citados anteriormente e pelo fato que a preferência do usuário exerce uma maior influência na tomada de decisão ao utilizar a teoria da utilidade, sendo assim, foi escolhido este método para o desenvolvimento do trabalho.

1.4 Teoria da Utilidade

1.4.1 Origem

Conforme apresentado por Cusinato (2003), Nicholas Bernoulli publicou o famoso Paradoxo de São Petersburgo, que enfatiza que "homens prudentes" nem sempre obedecem ao princípio da expectância matemática. Pode-se considerar, como um exemplo, um jogo cuja moeda é jogada repetidamente até aparecer a primeira "cara". É pago pelo jogo 2^{n-1} dólares se a primeira cara aparecer na enésima jogada. Assim, se o indivíduo levasse somente em consideração a expectância matemática, ele

estaria disposto a pagar, no máximo, o valor da mesma. Como o valor da esperança matemática é

$$E(L) = \sum_{n=1}^{\infty} \left(\frac{1}{2}\right)^{n} \cdot 2^{n-1}$$

O cálculo da mesma para este jogo seria:

$$E(L) = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 4 + \ldots + = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \ldots = +\infty$$

Portanto, ele pagaria qualquer preço para entrar neste jogo, não importando a riqueza ou não do mesmo, ele estaria disposto a entregar toda a sua riqueza para poder participar. O que certamente divergiria do comportamento observável no mundo real, no qual a maioria das pessoas não estaria disposta a dar muito dinheiro para isto.

Para o Paradoxo de São Petersburgo, Bernoulli propôs uma solução, argumentando que o valor que um indivíduo atribui a sua riqueza não corresponde ao próprio valor monetário desta, mas sim sua utilidade:

> "(...) a determinação do valor de um item não pode ser baseado em seu preço, mas sim na utilidade que ele fornece. O preço de um item depende somente do próprio item e é igual para todo mundo; a utilidade, contudo, depende das circunstâncias particulares do indivíduo que faz a estimativa." [Bernoulli, (1954) *apud* Cusinato, (2003)].

Desde a proposição que a "teoria da utilidade esperada" possa ser aplicada para modelar o comportamento dos tomadores de decisão, em 1944, por John Von Neumann e Oskar Morgenstern, diversos aprimoramentos e críticas surgiram. Leonard Savage, em 1954, por exemplo, propôs a inclusão de probabilidades subjetivas na ponderação das decisões futuras.

Para efetuar cálculos utilizando o princípio da expectância matemática, não necessitava fazer qualquer tipo de avaliação subjetiva, bastava somente multiplicar as probabilidades pelos possíveis resultados.

Porém, com a teoria da utilidade esperada, introduziu-se à teoria da decisão a subjetividade, isto é, a avaliação subjetiva dos tomadores de decisão passou a ter um papel fundamental. Os possíveis resultados e as probabilidades não eram mais suficientes para determinar a decisão, pois a utilidade varia com as circunstâncias de quem faz a estimativa.

1.4.2 Conceitos e terminologia

A teoria da utilidade é um método muito antigo, mas ainda muito utilizado. Assume que o decisor deseja fazer uma escolha que corresponda ao maior nível de satisfação ou utilidade, sendo que a mesma é relativa ao seu decisor.

Segundo Russell e Norvig (2004, p.570), as preferências do decisor são consideradas propriedades básicas, que atribui um único número para expressar a desejabilidade de um estado, isto é, dá um valor numérico maior para o melhor

resultado possível e um valor menor para um evento menos preferível. Esse método resolve algumas limitações da teoria da probabilidade.

As preferências expressas por utilidades, são combinadas com as probabilidades na teoria geral de decisões racionais, denominada de "teoria da decisão".

O método da Teoria da Utilidade será utilizado neste trabalho para auxiliar a construção da ferramenta proposta, para que seja possível representar e raciocinar com preferências.

1.4.3. Funcionamento da Teoria da Utilidade

De acordo com Jansen (2004, p. 2257), a operacionalização do método da teoria da utilidade é da seguinte maneira:

- Estruturação do problema em uma Matriz de Decisão, que é uma tabela onde são lançados nas colunas os elementos do problema como cenários, critérios, probabilidades, alternativas de decisão;
- Escolha do valor da utilidade, sendo que varia entre 0 e 1.

Atribui-se o valor 0 ao pior valor e o valor 1 ao melhor valor, tendo também os valores intermediários.

- Busca dos objetivos em cada uma das alternativas.
- A alternativa a ser escolhida será a que apresentar o maior valor de utilidade.

1.4.4 Vantagens e Desvantagens da Teoria da Utilidade

- **Vantagens**:

A Teoria da Utilidade permite maximizar a utilidade (preferência) esperada, permite também atribuir valores às preferências do usuário o que possibilita alcançar o melhor resultado esperado possível de acordo com a preferência do mesmo.

- **Desvantagens**:

O método da Teoria da Utilidade possui uma limitação, apresentada no momento de se realizar a análise de sensibilidade quando existem muitos critérios.

1.5 Teoria da Probabilidade

A probabilidade proporciona um meio para resumir a incerteza que vem de nossa preguiça e ignorância (RUSSELL e NORVIG, 2004).

1.5.1 Raciocínio Probabilístico

O raciocínio probabilístico é baseado na realização de inferências probabilísticas. O cálculo é baseado em probabilidade condicional (conceito primitivo) e no teorema de Bayes (descrito na Seção 1.5.2) sendo que a probabilidade condicional é vista como uma medida de crença no evento, dada todas as evidências disponíveis.

Para contornar os problemas acima, os primeiros sistemas computacionais para suporte à decisão eram fundamentados na teoria da decisão e tratavam incerteza com uma forma restritiva, mas viável em computador, da teoria probabilística de Bayes. Porém, o interesse no uso de probabilidades diminuiu devido à percepção (da época) de

que era intratável e inadequada para expressar a estrutura do conhecimento humano e porque em domínios maiores podiam produzir resultados matematicamente incorretos, além de não existirem mecanismos de explicação para os não especialistas do domínio.

Nos anos 70 surgem os sistemas especialistas, métodos derivados da teoria de probabilidade, que permitem a manipulação com êxito dos domínios de problemas maiores e mais complexos do que os anteriores, além de proverem facilidades de explicação, proporcionam o seu uso por não especialistas (LADEIRA, COELHO e VICCARI, 1998).

No final dos anos 80, houve uma retomada do interesse por abordagens probabilísticas motivada pela descoberta de que se considerando o relacionamento causal e a independência (condicional) entre variáveis do domínio, é necessário representar apenas probabilidades condicionais entre variáveis *diretamente* dependentes, tornando essa representação tratável em computador. Essa retomada está associada ao aparecimento de modelos baseados em representações gráficas de dependências probabilísticas denominadas *redes probabilísticas* (LADEIRA, COELHO e VICCARI, 1998). O uso dessas redes apresenta a vantagem, em relação às abordagens anteriores, de permitir a representação e manipulação da incerteza com base em princípios matemáticos fundamentados;

O raciocínio probabilístico em redes bayesianas é entendido como um processo de atualização de crenças (LADEIRA, COELHO e VICCARI, 1998).

1.5.2 Teorema de Bayes

A probabilidade a priori, também chamada de probabilidade incondicional, associada a uma hipótese *a* é o grau de crença (confiança) acordado para a hipótese na

ausência de quaisquer outras informações, representado por P(a). Porém, ao surgir alguma evidência relativa às variáveis aleatórias anteriormente desconhecidas, a probabilidade a priori não é mais útil. Em vez dela utiliza-se a probabilidade condicional ou posterior, representado por P(a|b), que é lida como "a probabilidade de a, dado que tudo o que se sabe é b". Um exemplo considerável é P(dordecabeçalvisão) = 0,6, onde uma pessoa tem dor de cabeça e não possui mais nenhuma informação sobre ela, a não ser a sua vista fraca.

Podem-se definir as probabilidades condicionais em termos de probabilidades incondicionais, pela equação:

$$P(a|b) = \frac{P(a \cap b)}{P(b)}, \quad P(b)>0. \tag{1}$$

$P(a \cap b) = P(a|b)P(b)$ esta equação é denominada regra do produto.

Esta equação, chamada regra do produto, pode ser escrita em duas formas:

$$P(a \cap b) = P(a|b)P(b)$$
$$P(a \cap b) = P(b|a)P(a)$$

Igualando as duas formas, pode-se chegar à regra de Bayes conhecida como Teoria de Bayes (Russell e Norvig, 2004):

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)} \tag{2}$$

Exemplo: Suponha que um grande número de caixas de bombons sejam compostas de dois tipos, A e B. O tipo A contém 70 por cento de bombons doces e 30 por cento de bombons amargos, enquanto no tipo B essas percentagens de sabor são inversas. Além disso, suponha-se que 60 por cento de todas as caixas de bombons sejam do tipo A, enquanto as restantes sejam do tipo B.

Cria-se o seguinte problema de decisão: uma caixa do tipo desconhecido é oferecida. Tendo a permissão para tirar uma amostra de bombom, com esta informação deve-se adivinhar se a caixa que foi oferecida é do tipo A ou se é do tipo B (MEYER, 1965).

Têm-se alguns cálculos:

$P(A) = 0,6$; $P(B) = 0,4$;
$P(S_d|A) = 0,7$; $P(S_d|B) = 0,3$;
$P(S_a|A) = 0,3$; $P(S_a|B) = 0,7$;

Suponha-se que realmente retirou-se um bombom de sabor doce. Qual decisão mais tenderia a ser tomada? Comparando:

$P(A| S_d)$ e $P(B| S_d)$.

Empregando-se a fórmula de Bayes, tem-se:

$$P(A| S_d) = \frac{P(S_d|A)P(A)}{P(S_d|A)P(A) + P(S_d|B)\,P(B)} =$$

$$\frac{(0,7)(0,6)}{(0,7)(0,6) + (0,3)(0,4)} = \frac{7}{9}.$$

O cálculo semelhante dará:

$P(B| S_d) = 2/9$.

Desta maneira, baseado na evidência obtida (isto é, a tirada de um bombom de sabor doce) é $2^{1/2}$ vezes mais provável que se esteja diante de uma caixa do tipo A, em vez de uma do tipo B (MEYER, 1965).

1.5.3 Redes Bayesianas

1.5.3.1 Conceitos e terminologia

Redes Bayesianas são grafos acíclicos direcionados, mostrando as relações de causalidade entre as variáveis. Nestes grafos, as elipses são as variáveis (atributos) e as ligações representam os relacionamentos de influência entre as variáveis. A partir dos cálculos estatísticos, cada variável terá uma tabela de valores de probabilidades para que suas possíveis ações sejam realizadas. Assim, na utilização de uma ferramenta de análise de Redes Bayesianas é possível definir hipóteses sobre uma determinada variável, tendo respostas sobre as influências por ela dadas de acordo com as ligações existentes entre as outras variáveis.

Russell e Norvig (2004) descrevem que uma rede bayesiana é um grafo orientado em que cada nó é identificado com informações de probabilidade quantitativa. A especificação completa é dada a seguir:

1. Um conjunto de variáveis aleatórias constitui os nós da rede. As variáveis podem ser discretas ou contínuas.

2. Um conjunto de vínculos orientados ou setas conecta pares de nós. Se houver uma seta do nó X até o nó Y, X será denominado pai de Y. Ocorre que o X tem uma influência direta sobre Y.

3. Cada nó X_i tem uma distribuição de probabilidade condicional $P(X_i|Pais(X_i))$ que quantifica o efeito dos pais sobre o nó.

4. O grafo não tem nenhum ciclo orientado (e consequentemente é um grafo acíclico orientado, ou GAO).

Para exemplificar a construção de uma Rede Bayesiana, pode-se analisar o seguinte exemplo: Tendo um novo alarme contra assaltantes instalado em uma casa. O alarme é bastante confiável na detecção de um roubo, mas também responde ocasionalmente a pequenos terremotos. Esta casa tem dois vizinhos, João e Maria, que prometeram chamar o dono da casa em seu trabalho quando ouvirem o alarme. João sempre chama quando houve o alarme, mas às vezes confunde o toque do telefone com o alarme e também liga ao ouvi-lo. Por outro lado, Maria gosta de ouvir música em alto volume e às vezes esquece completamente o alarme. Dada a evidência de quem telefonou ou não telefonou, para estimar a probabilidade de um roubo, é dada a rede bayesiana a seguir:

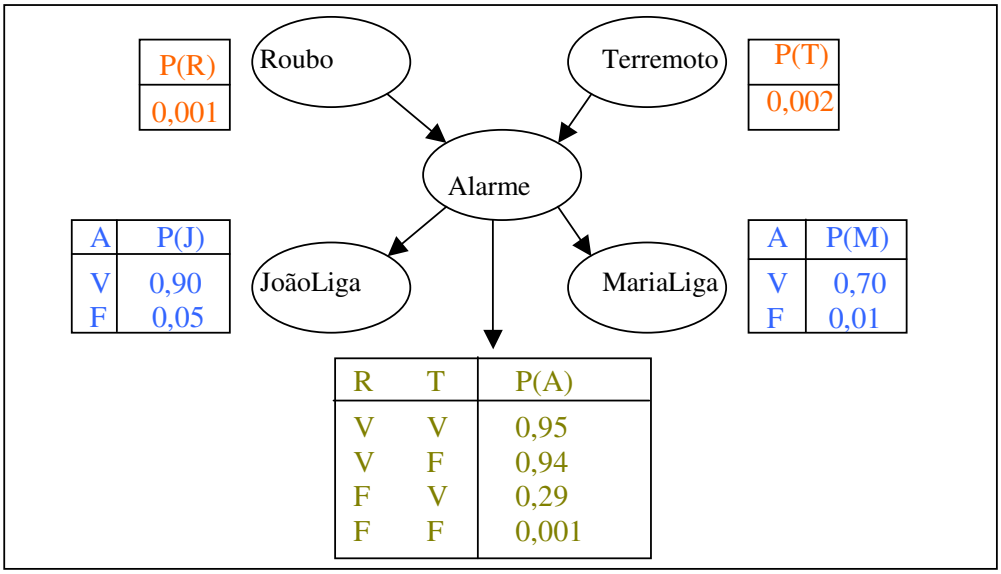


Figura 1.3 - Exemplo de Redes Bayesianas (Russell e Norvig, 2004).

Na Figura 1.3 pode-se observar que Roubo e Terremoto afetam diretamente a probabilidade de o alarme soar, mas o fato de João e Maria ligarem só depende do alarme.

Pode-se observar as distribuições condicionais nas tabelas de probabilidade condicional, ou TPC. Cada linha da TPC representa uma probabilidade condicional que cada nó possui para um caso de condicionamento. Um caso de condicionamento é apenas uma combinação possível de valores para os nós superiores. Para variáveis booleanas deve-se observar que se um valor verdadeiro é p, a probabilidade do valor falso é 1-p.

Nota-se que se uma variável possui K pais, portanto ela conterá 2^k probabilidades que podem ser especificadas de modo independente. E se não possuir nenhum pai a tabela terá apenas uma linha representando as probabilidades a priori de cada valor possível da variável.

- **Vantagens**

As vantagens da utilização das redes bayesianas segundo Ladeira et al (2000), concentram-se no sentido de aceitar a representação e manipulação da incerteza com base em princípios matemáticos fundamentados, e ainda, modela o conhecimento do especialista do domínio de uma forma intuitiva.

Uma outra vantagem do uso de redes bayesianas é sobre as representações de incerteza utilizando a Teoria da Probabilidade, pois somente esta teoria fornece consistência para fazer tais interpretações e, portanto pode ser usada em sistemas de apoio à decisão. Além disso, é o único formalismo que permite realizar qualquer um dos

tipos possíveis de inferência probabilística, ou seja, causal, diagnóstico, intercausal ou misto.

As redes bayesianas tornam ainda o processo de inferência eficiente computacionalmente (SILVA e LADEIRA, 2002). As redes também admitem analisar grandes quantidades de dados, para extrair conhecimentos úteis em tomada de decisões, controlar ou prever o comportamento de um sistema, diagnosticar as causas de um fenômeno entre outros.

As redes bayesianas modelam a obtenção de conhecimentos, permite juntar e fusionar conhecimentos de naturezas diferentes num mesmo modelo como dados históricos ou baseado na experiência, experiência expressa na forma de regras lógicas, de equações, de estatísticas ou de probabilidades subjetivas. Por exemplo, no mundo industrial, cada uma das fontes de informação, embora presente, é repetidamente insuficiente para fornecer uma representação precisa e realista do sistema analisado.

- **Desvantagens**

> "os métodos bayesianos requerem uma significativa quantidade de dados probabilísticos para construir uma base de conhecimento; frequentemente o tipo de relacionamento entre a hipótese e a evidência é importante na determinação de como a incerteza será gerenciada; a redução das associações para números também elimina o uso desse conhecimento dentro do limite de outras tarefas" (GONZALEZ e DOUGLAS, 1993).

1.5.3.2 Semântica das redes bayesianas

Há duas maneiras de se compreender a semântica das redes bayesianas (RUSSELL e NORVIG, 2004):

- enxergar a rede como uma representação da distribuição de probabilidade conjunta;
- visualizar a rede como uma codificação de uma coleção de declarações de independência condicional.

➢ **Representação da distribuição conjunta total**

A rede bayesiana fornece uma descrição completa do domínio, porque toda a entrada que ocorrer na distribuição de probabilidade conjunta total pode ser calculada a partir das informações que estarão armazenadas na rede. Isto pode ser representado por meio da equação $P(X_1 = x_1 \wedge x_2 \wedge x_3 \wedge ... \wedge X_n = x_n)$ onde X é a variável e x são as atribuições. Usou-se a fórmula $P(x_1,...,x_n)$ como uma simplificação para isto. O valor dessa entrada é dado pela fórmula.

$$p(x_1,\ldots,x_n) = \prod_{i=1}^{n} p(x_i \mid pais(x_i)) \quad (3)$$

Os pais (X_i) representam os valores que são simulados na tabela filho. Assim, cada entrada na distribuição conjunta é representada pelo produto dos elementos apropriados das tabelas de probabilidade condicional (TPCs) na rede bayesiana.

Abaixo será ilustrado um exemplo elaborado por Russell e Norvig (2004):

Para calcular a probabilidade de que o alarme tenha soado, mas não tenha ocorrido nenhum roubo nem um terremoto, e que tanto João quanto Maria tenham ligado, foi usado nomes de uma única letra para identificar as variáveis:

$P(j \wedge m \wedge a \wedge r \wedge \neg t)$
$= P(j|a) * P(m|a) * P(a|\neg r \wedge \neg t) * P(\neg r) * P(\neg t)$
$= 0,90 * 0,70 * 0,001 * 0,999 * 0,998 = 0,00062.$

P(j|a) - João ligou;
P(m|a) - Maria ligou;
P(a|¬r^¬t) - não ocorreu nem roubo e nem terremoto, mas o alarme disparou;
P(¬r) - não ocorreu roubo;
P(¬t) - não ocorreu terremoto;

- **Método para construir redes bayesianas**

Para construir uma rede Bayesiana com a estrutura correta para o domínio é necessário escolher os pais para cada nó de forma que os pais de um nó x_i incluam todos os nós $x_1,\ldots, x_i\text{-}1$ que influenciam diretamente x_i;

Métodos utilizados:

- Escolher o conjunto de variáveis relevantes x_i que descrevem o domínio;
- Selecionar uma variável x_i e adicionar um nó à rede para ela;
- Definir os *pais* (x_i) como o conjunto mínimo de nós já existentes na rede para os quais a propriedade de independência condicionada se verifique;
- Definir a tabela de probabilidade condicionada para x_i;

Uma vez que cada nó apenas se liga a nós definidos anteriormente, este método de construção garante que a rede é acíclica.

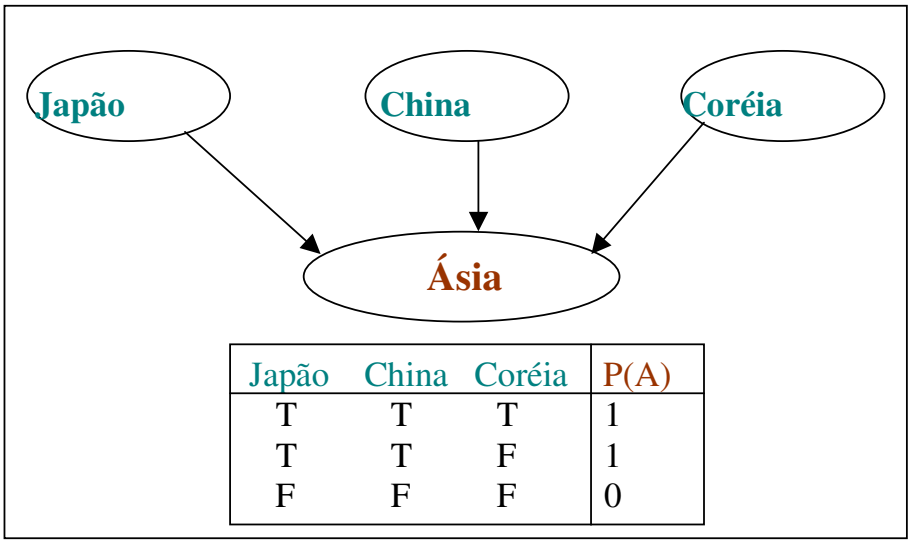
- **Representação eficiente de distribuições condicionais**

Muitas vezes o preenchimento das tabelas de probabilidades condicionadas é simples desde que a relação entre os pais e o nó filho não seja arbitrária.

O exemplo mais simples é fornecido por *nós determinísticos* que têm o seu valor especificado a partir dos valores dos pais, sem qualquer incerteza (p=1, ou p=0).

O relacionamento pode ser:

– Relação lógica: Exemplo: disjunção, como mostrada na Figura 1.4, onde três variáveis com situações diferentes interferem no resultado de uma variável.



Japão	China	Coréia	P(A)
T	T	T	1
T	T	F	1
F	F	F	0

Figura 1.4 - Interferência em Variável

– relação numérica. Exemplo:

- Pais: preço de um dado modelo de carro em vários stands;

- Filho: Preço negociado; o valor é o mínimo dos valores dos pais, sem incerteza;

- **Inferência usando Redes Bayesianas**

O serviço que a inferência probabilística faz é calcular a distribuição de probabilidade posterior para um conjunto de variáveis de consulta, dado algum evento observado isto é, alguma atribuição de valores a um conjunto de variáveis de evidência.

A distribuição de probabilidade conjunta pode ser usada para responder a qualquer pergunta sobre o domínio. As redes Bayesianas, que são as representações gráficas das distribuições, podem também ser usadas para responder a qualquer questão de inferência do tipo:

- diagnóstico (dos efeitos para as causas)
- P(Assalto|JoãoLiga) = 0.019
- causais (das causas para os efeitos)
- P(JoãoLiga| Assalto) = 0.73
- intercausais (entre causas e um efeito comum)
- P(Assalto|Alarme) = 0.469
- P(Assalto|Alarme,Terremoto) = 0.001
- mistas (combinação das anteriores)

1.6 Considerações Finais

Este capítulo apresentou os conceitos básicos necessários para o entendimento e valorização desta monografia. As definições de mineração de dados, teoria da utilidade e teoria da probabilidade ajudam o leitor para um melhor entendimento da construção da ferramenta desenvolvida neste trabalho e suas funcionalidades, permitindo ao mesmo uma visão geral de como as teorias citadas foram aplicadas. Para o desenvolvimento da ferramenta proposta nesta monografia foi preciso a utilização de tecnologias, conforme citado no Capítulo 2.

2. TECNOLOGIAS UTILIZADAS

2.1 Banco de Dados e SGBD

Banco de Dados (BD) é um repositório de dados, onde os dados relevantes de uma aplicação são armazenados, que de acordo com Elmasri e Navathe (2005. p.04), pode ser criado e mantido manualmente, como por exemplo, um catálogo de cartões de funcionários de uma determinada empresa, ou automatizado por um grupo de aplicativos escritos especialmente para essa função ou por um SGBD, por exemplo.

Em um Banco de Dados, é possível manipular os dados pelas operações de inserção (*insert*), remoção (*delete*), atualização (*update*), recuperação (*select*) dos dados das tabelas existentes e adição de arquivos ao Banco de Dados (*createdb*).

Segundo Silberschatz, Korth e Sudarshan (1999. p.04), um Sistema Gerenciador de Banco de Dados (SGBD) "é uma coleção de arquivos e programas inter-relacionados que permite ao usuário o acesso para consultas e alterações desses dados", o uso do mesmo proporciona um ambiente conveniente e eficiente para armazenamento e recuperação das informações do BD, tendo como principal vantagem oferecer ao usuário uma visão abstrata dos dados, isto é, esconder dele determinados detalhes da forma de armazenamento e manutenção desses dados.

O SGBD escolhido para a manipulação dos dados desta ferramenta foi o PostgreSQL. Tal escolha foi feita tendo em vista que o mesmo é um SGBD livre (*open source*), isto é, sua licença não impõe restrições quanto ao seu uso, por oferecer funcionalidades necessárias para o desenvolvimento da ferramenta, como a herança, o polimorfismo, que são orientadas a objetos, e ter comandos específicos para buscar

tabelas que não possuem chave primária (consideradas como tabelas que possuem interação) (Ver APÊNDICE C.1 - Codificação).

O PostgreSQL foi desenvolvido pela Universidade de Berkley e atualmente é mantido pela imensa comunidade de desenvolvedores do mundo. É compatível com a maioria dos sistemas operacionais, incluindo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e Windows, que o foi o sistema operacional utilizado para este trabalho.

2.2 JDBC (*Java Database Conectivity*)

A interface JDBC é uma API para execução de SQL em qualquer Banco de Dados e também permite aos usuários escrever aplicações de BD usando uma interface puramente Java. A Sun oferece uma maneira gratuita de acesso a BD a partir de Java pela ponte JDBC-ODBC, o que significa que a API JDBC usa o ODBC para ter acesso ao Banco de Dados.

O PostgreSQL, que é o Banco de Dados usado neste trabalho, fornece um driver JDBC *tipo 4*, o que indica que o mesmo é escrito inteiramente em Java e se comunica utilizando o protocolo de rede nativo do sistema de Banco de Dados. Por esse motivo, ele é independente da plataforma, ou melhor, uma vez compilado, o driver pode ser utilizado em qualquer sistema.

2.3 Linguagem de programação

A linguagem de programação escolhida para o desenvolvimento da Ferramenta foi Java. Tal decisão foi devida a sua característica multiplataforma, sendo possível rodar em qualquer ambiente, por sua interatividade com Banco de Dados, por possuir API's eficazes e completas, além de sua aceitação no mercado.

3. FUNCIONALIDADES DA FERRAMENTA

3.1 Ferramenta DaMiTeCa

A ferramenta desenvolvida nesta monografia, denominada DaMiTeCa, foi construída na linguagem de programação Java utilizando o Eclipse 3.1 como ambiente de desenvolvimento e o SGBD PostgreSQL.

3.1.1 Objetivos

A partir dos dados que são gerados pela Ferramenta PLAVIS, que tem como objetivo gerar casos de testes em relação a um número de mutantes verificando o estado do mesmo (morto, vivo, em execução ou se ainda não foi executado), e são armazenados no *schema* da base de dados PLAVIS, a ferramenta desenvolvida neste trabalho, extrai os dados selecionados pelo usuário da base de dados PLAVIS e estes são armazenados na base de dados da ferramenta deste projeto.

Esta extração faz parte do processo de pré-processamento da mineração de dados, utilizando a técnica de transformação de dados, conforme citado no Capítulo 2. Após a extração, os dados são armazenados em uma base de dados que foi considerado

como sendo um *data warehouse* que tem como finalidade armazenar os dados extraídos por meio de uma seleção.

Os dados que estão contidos neste *data warehouse* servem como entrada para o algoritmo de redes bayesianas para que através dele possa ser extraído um elevado número de informação. Com base nas preferências do usuário, (por exemplo, se ele quer somente os mutantes vivos) e a utilidade desejada, a ferramenta verifica quais os casos de testes que são mais ágeis sobre o número de mutantes escolhido pelo usuário, mostrando ao mesmo os casos de testes que possuem um melhor resultado em relação à escolha atual, sendo que este resultado pode variar com a mudança de escolhas.

3.1.2 Arquitetura

A Ferramenta DaMiTeCa é composta por 3 módulos dependentes um do outro, pois o módulo posterior necessita dos resultados do módulo anterior. Tais módulos, ilustrados na Figura 3.1, são: Módulo de Busca dos Dados, Módulo de Análise dos Dados e Módulo de Visualização.

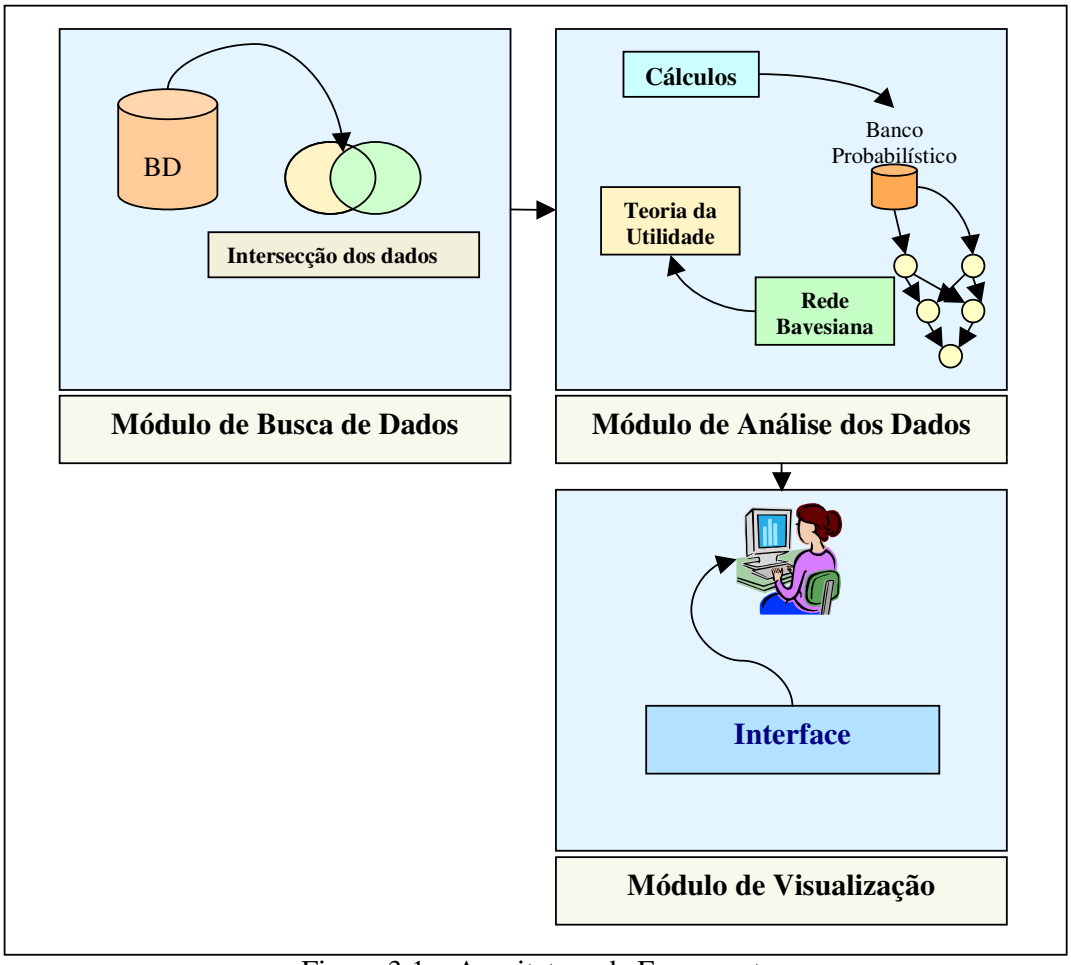


Figura 3.1 – Arquitetura da Ferramenta

A primeira etapa da Ferramenta, Módulo de Busca dos Dados, é responsável pela extração dos dados da base original, fazendo a intersecção dos dados escolhidos pelo decisor para que os mesmos possam ser armazenados no Banco de Probabilidade do módulo seguinte.

Na segunda etapa, Módulo de Análise dos Dados, por meio de técnica de mineração de dados, os dados que foram submetidos a intersecções são minerados, escolhidos, para, finalmente serem armazenados no Banco de Probabilidade e serem subordinados a cálculos que também são armazenados no Banco de Probabilidade. Com estes é feita uma rede bayesiana que traz a informação necessária para a aplicação da Teoria da Utilidade.

Por fim, a última etapa, Módulo de Visualização, resultado obtido e os principais dados que levaram a este resultado são mostrados ao decisor de uma forma clara e objetiva.

3.2 Interface

A conexão com o Banco de Dados é obrigatoriamente o primeiro passo da interação com o sistema, sendo estabelecida por meio do botão “Conectar BD”.

Na Figura 3.2 é ilustrada a tela principal do sistema, na qual o usuário digita o nome do Banco de Dados, o nome do usuário e a senha para que em seguida possa ser passado para a segunda etapa, que é a de selecionar a tabela, atributos e dados.

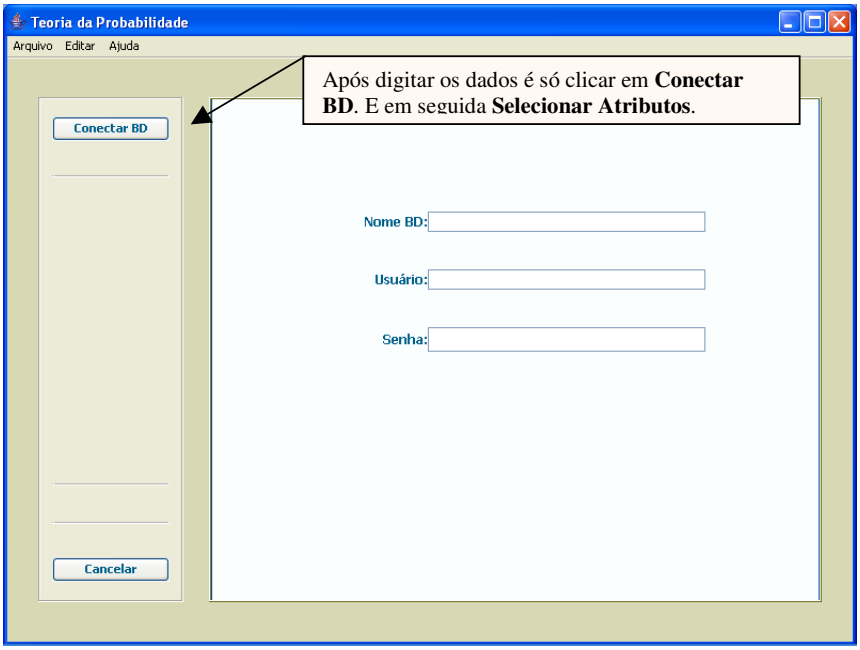


Figura 3.2 - Tela Principal da Ferramenta

Na Figura 3.3 é ilustrada a tela na qual o usuário informa a tabela que deseja extrair informações, e ao clicar em “Selecionar Tabela” o sistema informará os atributos que compõem a tabela, sendo estes os que ele poderá selecionar para que o sistema

possa fazer uma busca nas opções possíveis de extração de informação. Os dados são obtidos por meio de comandos que fazem buscas em forma de agrupamento, somente identificando as opções que são inseridas no Banco de Dados.

Em seguida, deve selecionar “Atributos” que o programa informará os dados que contém em cada atributo permitindo que o usuário selecione-os de acordo com a sua necessidade.

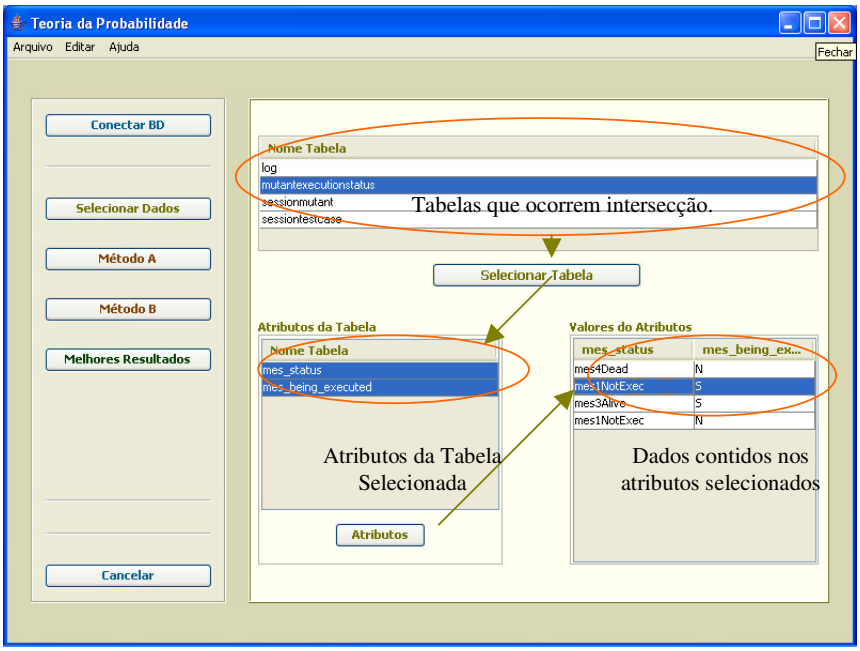


Figura 3.3 - Tela Selecionar Atributos

Após a definição, feita pelo usuário, das tabelas, atributos e dados, o sistema permitirá que o usuário escolha qual o método que é mais adequado para a informação que deseja obter, tendo como opção o Método A, que representa a probabilidade a priori ou probabilidade incondicional e o método B, representando as probabilidades condicionais ou posteriores e, que serão mais detalhadamente explicado na Seção 2.2. A partir do método escolhido será utilizado os conceitos da teoria da utilidade para a obtenção da melhor decisão de acordo com as preferências do usuário.

Com o método escolhido, o sistema informará o(s) dado(s) que foi selecionado e permitirá que o usuário selecione os caminhos que deseja percorrer para chegar até a tabela selecionada anteriormente, que é umas das três tabelas que ocorrem intersecções dos dados, mostrado na Figura 3.4. Para melhor rapidez do uso da ferramenta, o usuário pode buscar apenas os dados desejados no campo “Digite o dado:” da Figura 3.3, ou pode buscar todos os dados apenas deixando o mesmo em branco.

Assim, o sistema fará várias buscas para extrair as informações que o usuário necessita.

Na Figura 3.4, considera-se que o usuário selecionou a tabela User_, em seguida, a tabela Operator e após a intersecção feita pelo sistema, o usuário selecionou a tabela Mutant, retornando a intersecção para a tabela final, que no caso é a MutantExecutionStatus.

Ao chegar à tabela que tem como finalidade retornar os resultados que o usuário deseja, a tabela MutantExecutionStatus, o sistema informará ao mesmo que os dados já foram selecionados e perguntará se deseja criar um outro caminho. Caso a resposta seja positiva, o sistema pedirá que informe um novo caminho, senão pedirá para clicar em um dos botões: “Mostrar Resultado”, “Tabelas Resultados” ou “Distribuição Total”. Estas informações são ilustradas na tela do sistema, na Figura 3.4.

Caso o método escolhido seja o método do tipo B a escolha de n caminhos resultaria em várias intersecções na última tabela, que é a selecionada pelo usuário (Ver APÊNDICE A – Intersecções dos dados).

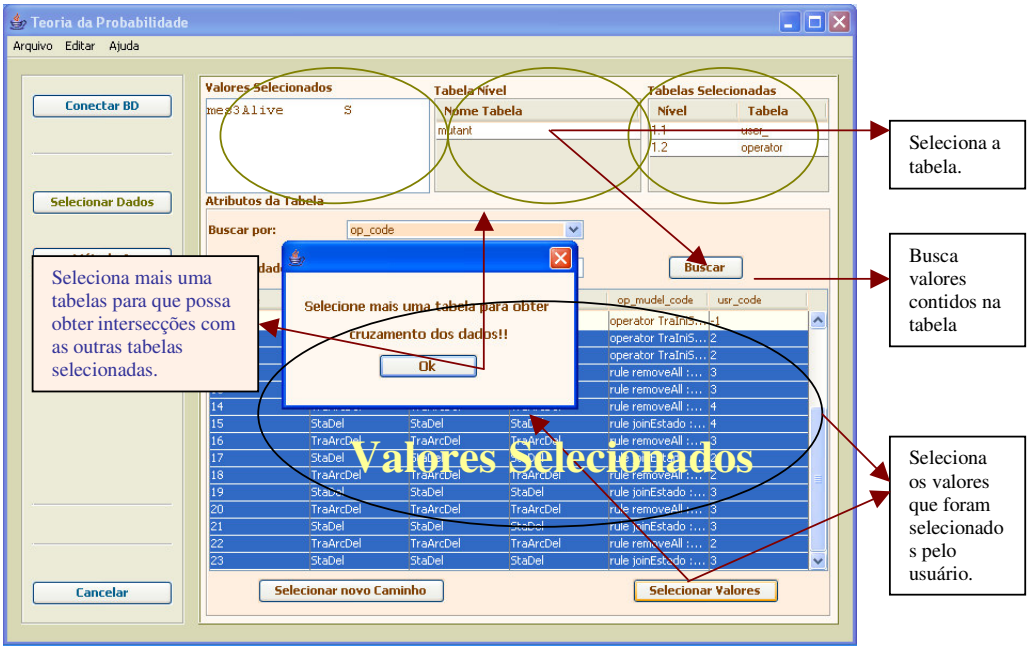


Figura 3.4 – Tela dos Métodos

Ao término do caminho escolhido pelo usuário, será informado ao mesmo que o caminho foi concluído, conforme é ilustrada na Figura 3.5, porém, será possível selecionar outro caminho, caso o usuário ainda o queira.

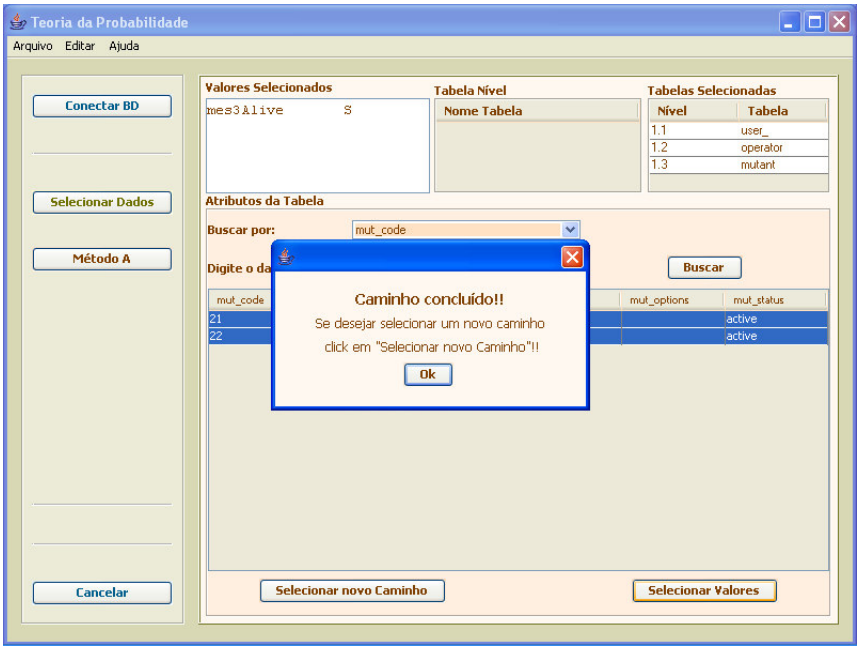


Figura 3.5 - Tela de Caminho Concluído

Na Figura 3.5 é apresentada a tela da opção “Mostrar Resultados” que informa os resultados que foram obtidos. Caso o usuário deseje saber quais foram os valores selecionados, deverá clicar em cada campo que o sistema informará na tabela abaixo.

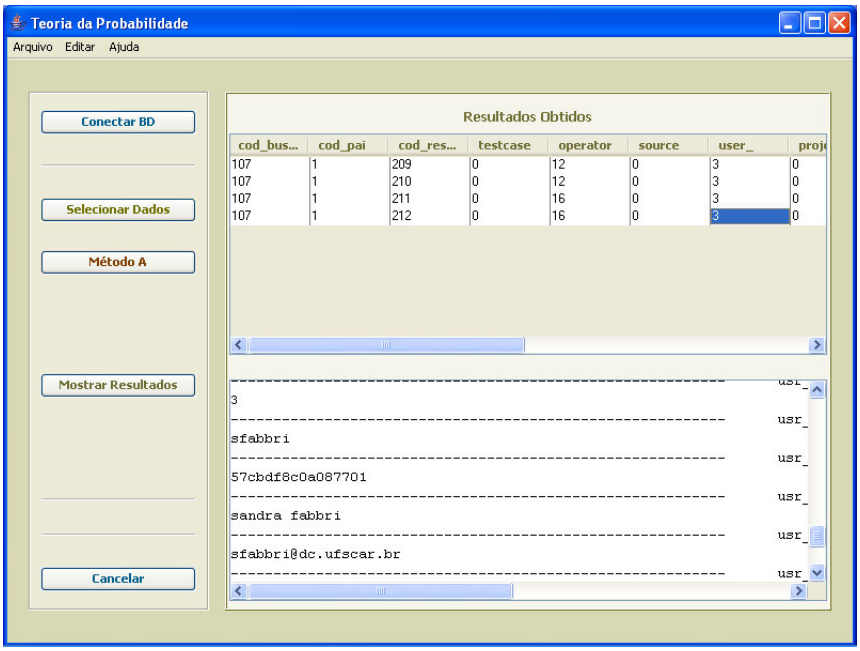


Figura 3.6 – Tela da opção “Mostrar Resultados”

Na Figura 3.7 é mostrada a tela da opção “Distribuição Total” que informa as tabelas selecionadas e os valores contidos em cada tupla. Os resultados dos cálculos são informados à medida que forem processados os dados.

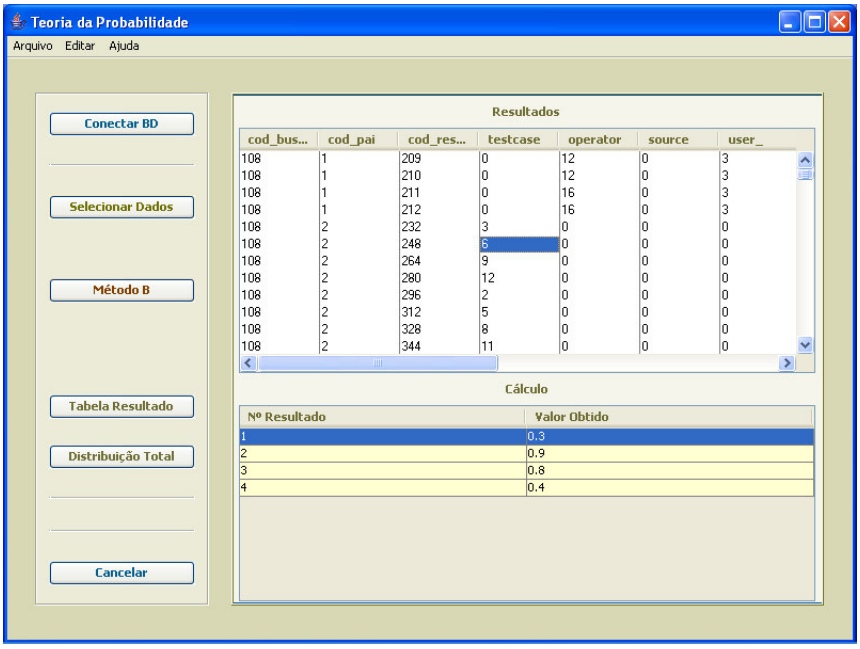


Figura 3.7 – Tela da opção “Distribuição Total”

Na Figura 3.8 é mostrada a tela que retorna a “Tabela Resultado” informando os valores de entrada e seus respectivos resultados obtidos, incluindo as probabilidades calculadas pelo sistema. Na segunda tabela são informados os dados contidos em cada coluna da tabela anterior.

Caso o usuário deseje saber quais foram os valores selecionados, deve somente clicar em cada campo da segunda tabela que o sistema informará na terceira tabela os resultados.

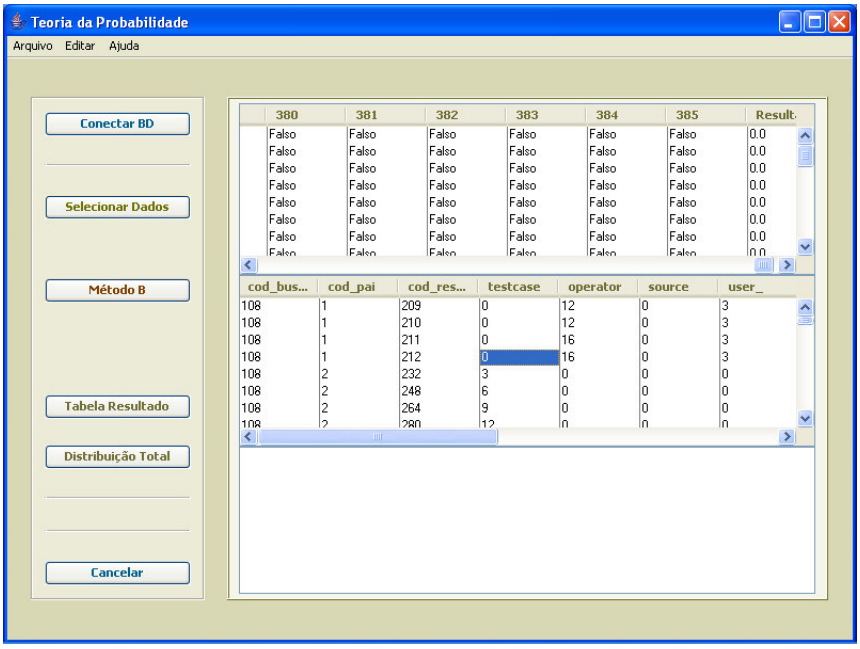


Figura 3.8 – Tela “Tabela Resultado”

3.3 Extração de Informação

3.3.1 Esquema da Base de Dados - PLAVIS

Na Figura 3.9 abaixo ilustra o projeto da base de dados que foi utilizada para desenvolver esta ferramenta.

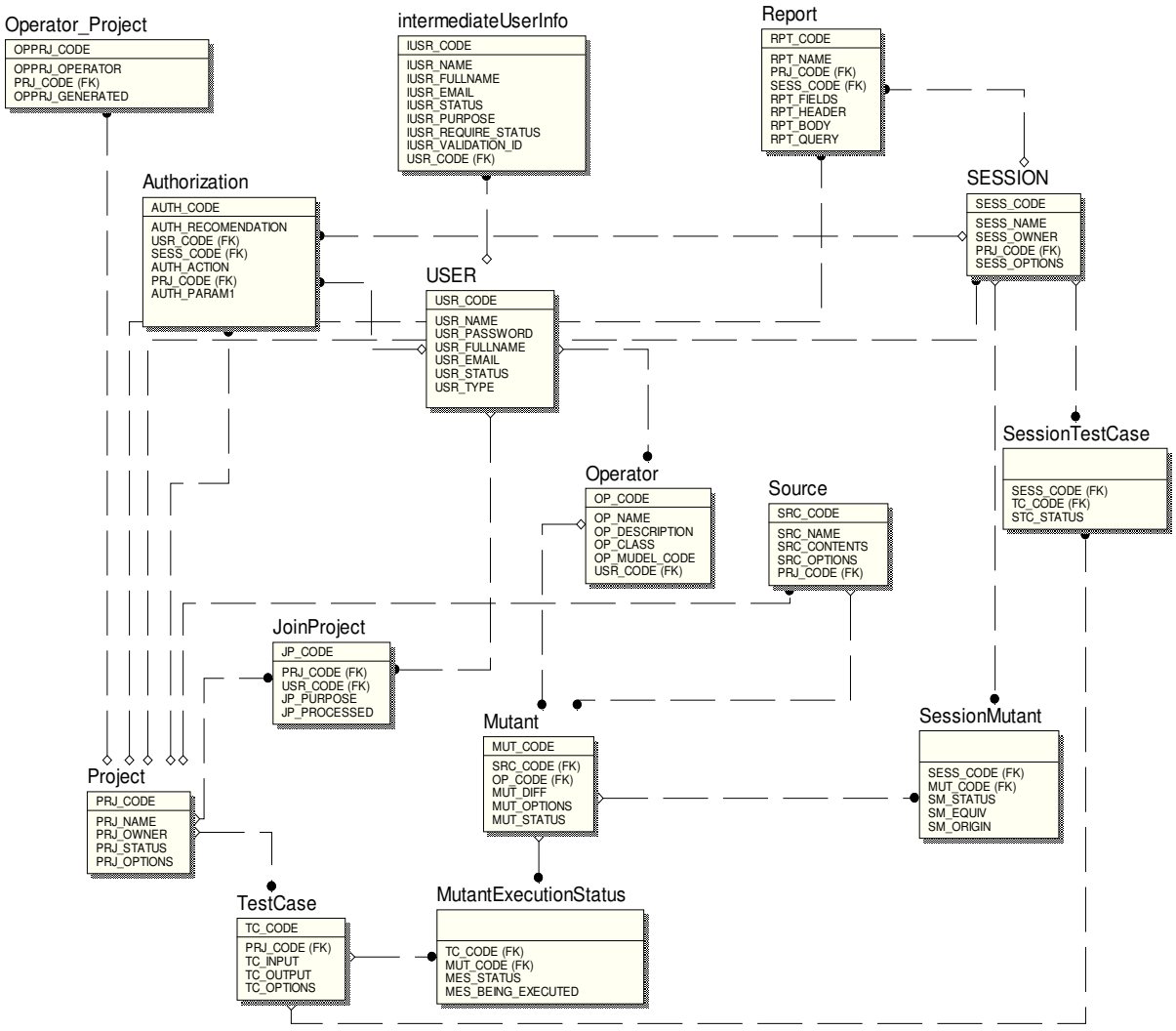


Figura 3.9 – Esquema PLAVIS

A Ferramenta PLAVIS do INPE (Instituto Nacional de Pesquisas Espaciais) tem como finalidade cadastrar dado para que possam ser gerados casos de testes em relação aos mutantes.

3.3.2 Modelo Conceito

O diagrama de classes apresentado na Figura 3.10 abaixo contém as classes do sistema desenvolvido. O sistema utilizou seis classes:

- *DM_Conexao:* contém método para se conectar com o Banco de Dados;
- *DM_Principal:* instancia as classes DM_Resultado_TPC, DM_Resultado_MetodoB, DM_Resultado_MetodoA e DM_BuscaTabelaMetodosDados para que possam realizar as operações, de acordo com a necessidade do usuário;
- *DM_Resultado_TPC:* contêm métodos que tem como objetivos construir as tabelas de probabilidades;
- *DM_Resultado_MetodoB:* possui métodos que retornam os cálculos obtidos pelas intersecções das tabelas utilizando a Representação da distribuição conjunta total.
- *DM_Resultado_MetodoA:* possui métodos que buscam os resultados obtidos pela intersecção das tabelas a partir da probabilidade a priori que foi descrita nos capítulos anteriores;
- *DM_select:* possui métodos que contêm os comandos SQL utilizados para realizar buscas e inserções e também tem atributos globais que caminham por todas as classes informando dados que sejam necessários para o retorno das informações obtidas;
- *DM_BuscaTabelaMetodosDados:* contêm métodos que são utilizados para retornar dados, para que o usuário possa selecionar e realizar as buscas que deseja.
- *DM_SelecionarTabelasDados:* contêm métodos que realizam as intersecções das tabelas para que possa realizar os cálculos probabilísticos.

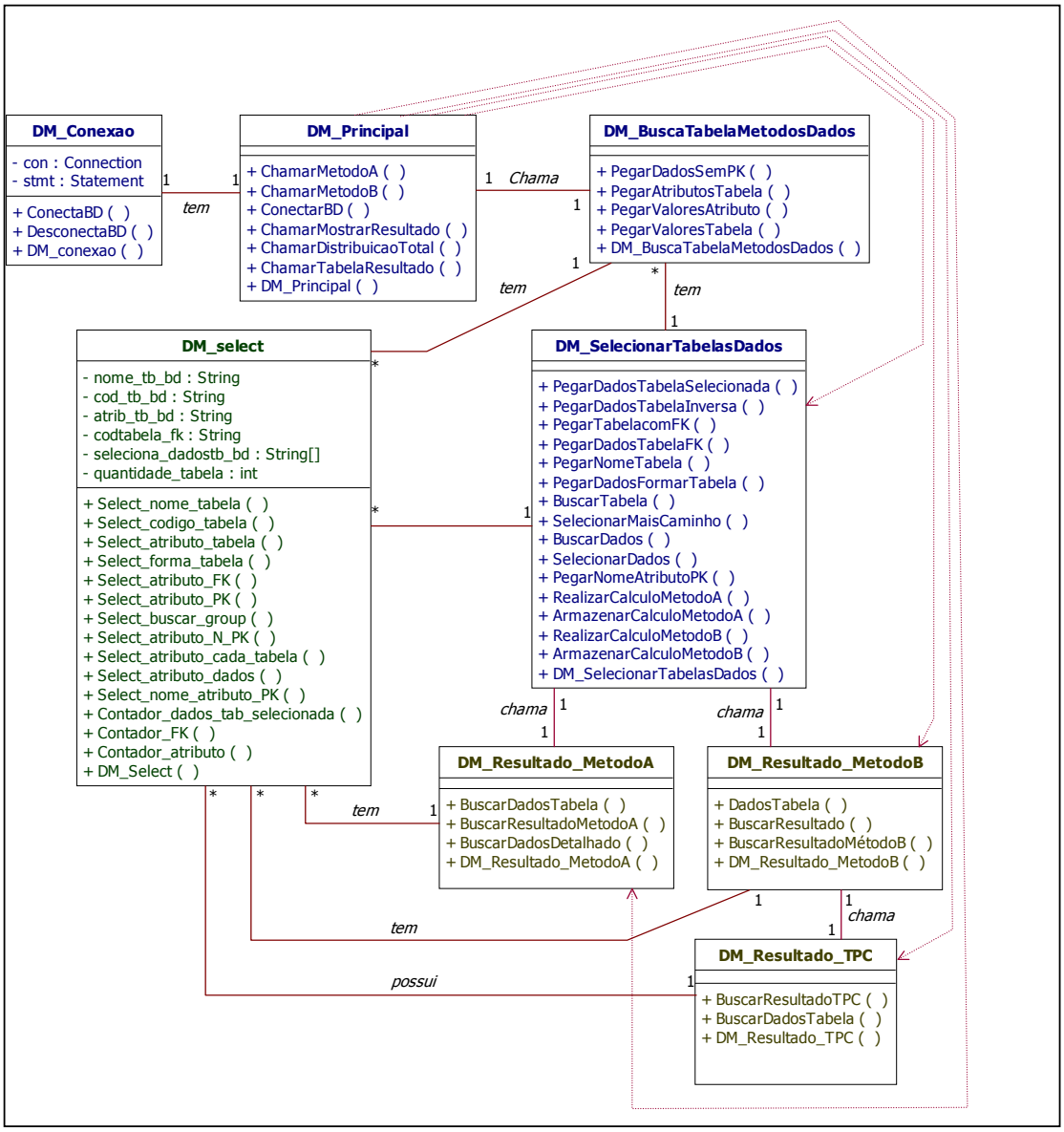


Figura 3.10 – Diagrama de Classe do Sistema

3.4 Armazenamento dos Dados Pré-Processados

O autor Elmasri e Navathe (2005) descreve sete considerações que um projeto de pré-processamento deve seguir, mas neste projeto só foram considerados três:

- *Projeções de uso:* descreve o modo de como o usuário interage com o sistema e como fazer;

- *O ajuste do modelo de dados:* escolha do modelo de dados para dar suporte ao uso;
- *Projeto do componente de metadados (metadados):* descrição de um Banco de Dados incluindo sua definição de esquema.

Na Figura 3.11 é ilustrado o projeto de transformação dos dados onde os dados brutos são transformados e colocados em outra base de dados.

Figura 3.11 – Projeto de Transformação dos Dados

Para o armazenamento, dos dados pré-processados, foi criado mais um *Schema* na mesma base de dado do PLAVIS. O novo *Schema* foi denominado de *data warehouse* por possuir as mesmas características de uma *Data Warehouse* que tem como finalidade armazenar dados pré-processados. A transformação do dado que foi utilizada foi citada no capítulo 2.

As tabelas são identificadas no Banco de Dados e criadas automaticamente de acordo com a necessidade (Ver APÊNDICE C.2 - Codificação). São identificadas por não possuir chave primária, pois são tabelas que ocorre intersecção das outras tabelas, ressaltando que o comando que faz esta busca foi citado anteriormente, tendo também o código para a criação da tabela no *Schema* datawarehouse (Ver APÊNDICE C.3 - Codificação).

3.5 Cálculos Probabilísticos

Na Figura 3.12 é ilustrada uma das etapas do sistema. Tem como finalidade extrair os dados da base de dados, que contém o pré-processamento e os cálculos obtidos pela teoria da probabilidade, para serem utilizados como entrada para a rede bayesiana. Logo abaixo conterá uma explicação detalhada do sistema.

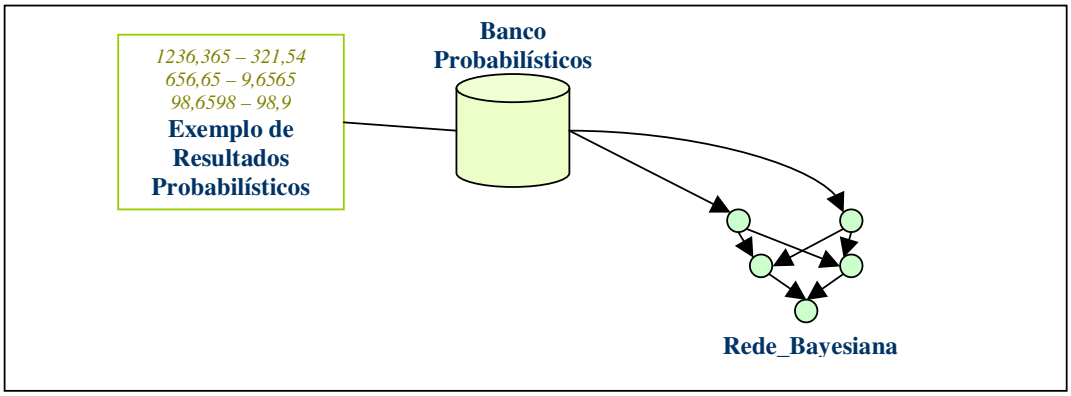


Figura 3.12 - Principais Etapas do Sistema

Elmasri e Navathe (2005) afirmam que não há uma maneira de antecipar todas as possíveis consultas ou análises durante a fase de projetos, mas o projeto deveria dar suporte especificamente a acesso aos dados em qualquer combinação significativa dos valores para os atributos nas tabelas.

A partir dos conceitos de Elmasri e Navathe, o sistema foi desenvolvido para uma maior interação com o usuário, possibilitando que ele busque os resultados de acordo com as suas necessidades.

Para a realização destas buscas foi necessária utilização das bibliotecas específicas do PostgreSQL, que possui comandos que realizam operações avançadas na base de dados, buscando identificar detalhadamente as tabelas existentes na base de dados e, em seguida, seus respectivos atributos. E também foi utilizado para verificar as ligações existentes entre as tabelas identificando suas chaves primária e secundária. Isto

foi necessário para que independente de qual for o nome da tabela e seus atributos, as consultas sejam realizadas da forma que o usuário necessitar.

A ferramenta foi desenvolvida com o objetivo de extrair um número significativo de informação da Ferramenta PLAVIS que tem como foco principal gerar vários casos de teste em cima de um número de mutantes para descobrir quais destes casos de teste podem ter matado o mutante. Os mutantes são considerados trechos que ocorrem erros e que podem ser descobertos com ataques dos casos de teste que possui características diferentes um do outro.

As tabelas MutantExecutionStatus, SessionMutant e SessionTestCase são responsáveis por armazenar as interações dos cruzamentos das outras tabelas. Estas tabelas armazenam um grande volume histórico decorrente do processo de interações que ocorreram na execução dos testes.

Na Figura 3.13 são ilustradas as tabelas que guardam as informações. A tabela Mutant armazena os principais atributos, a tabela TestCase guarda os casos de testes que agem sobre o mutante e, por fim, a tabela MutantExecutionStatus que tem por finalidade guardar a interação que ocorre entre as duas tabelas a TestCase e a Mutant (Ver APÊNDICE C.4 - Codificação).

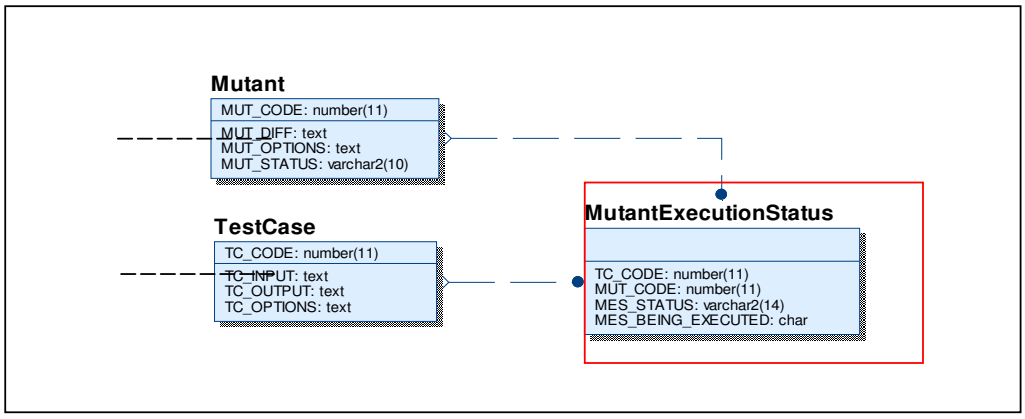


Figura 3.13 – Representação da Tabela MutantExecutionStatus

A segunda interação ocorre está ilustrada na Figura 3.14 que possui as tabelas Session, Mutant e SessionMutant.

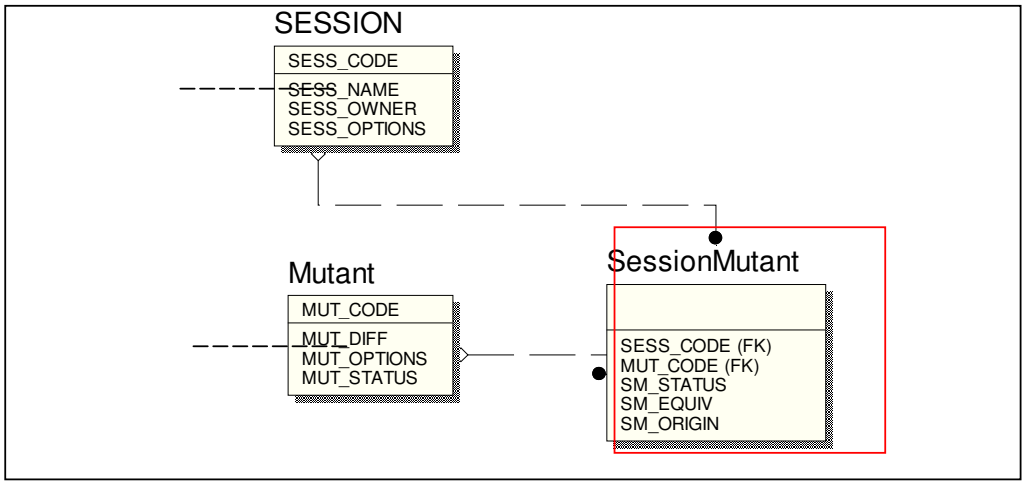


Figura 3.14 – Representação da Tabela SessionMutant

A terceira interação está representada na Figura 3.15, que possui as tabelas Session, TestCase e SessionTestCase.

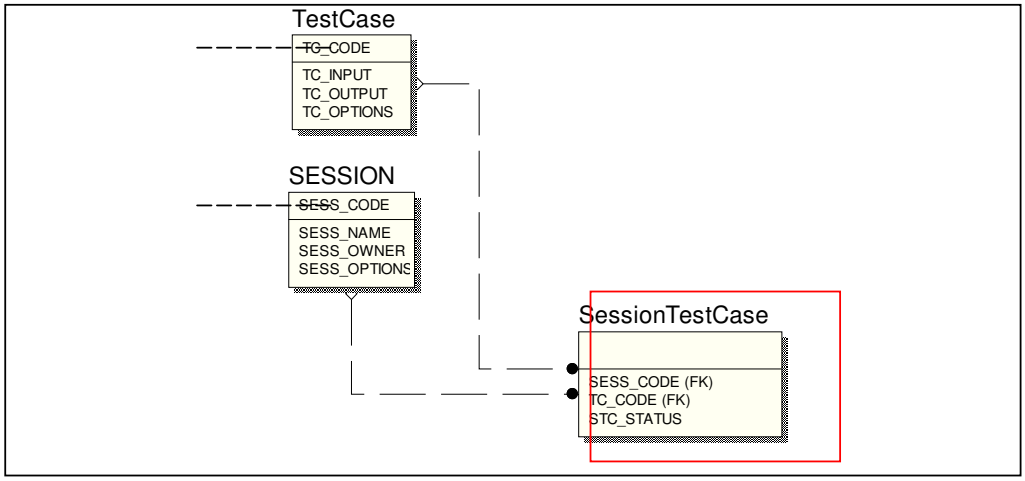


Figura 3.15 – Representação da Tabela SessionTestCase

Com a análise das três tabelas, conclui-se que as tabelas que não possuem chave primária podem ser consideradas como sendo as tabelas que possuem interação, tabela de cruzamento das outras tabelas. Portanto, foi utilizado o comando do Banco de Dados PostgreSQL que busca somente estas tabelas. (Ver APÊNDICE C.1 - Codificação).

Em seguida utilizou-se um comando SQL (Ver APÊNDICE C.5 - Codificação) que tem como finalidade retornar o código da tabela que será utilizado para retornar os atributos que não são nem chave primária nem secundária.

A partir disso, foi definido quais são os atributos que não fazem parte de nenhuma outra tabela, não são chaves secundárias nem primárias (Ver APÊNDICE C.6 - Codificação).

3.6 Métodos A e B da Ferramenta DaMiTeCa

- **Método A**

O Método A representa a probabilidade a priori ou probabilidade incondicional que tem como finalidade associar uma hipótese *a* na ausência de qualquer outra informação representado como P(a). Um exemplo que pode ser considerado seria P(MutanteVivo) = 0,7, no qual informa a quantidade de mutante vivo que existe armazenado na tabela de cruzamento MutantExecutionStatus. Na Figura 3.16 são apresentados os resultados deste método.

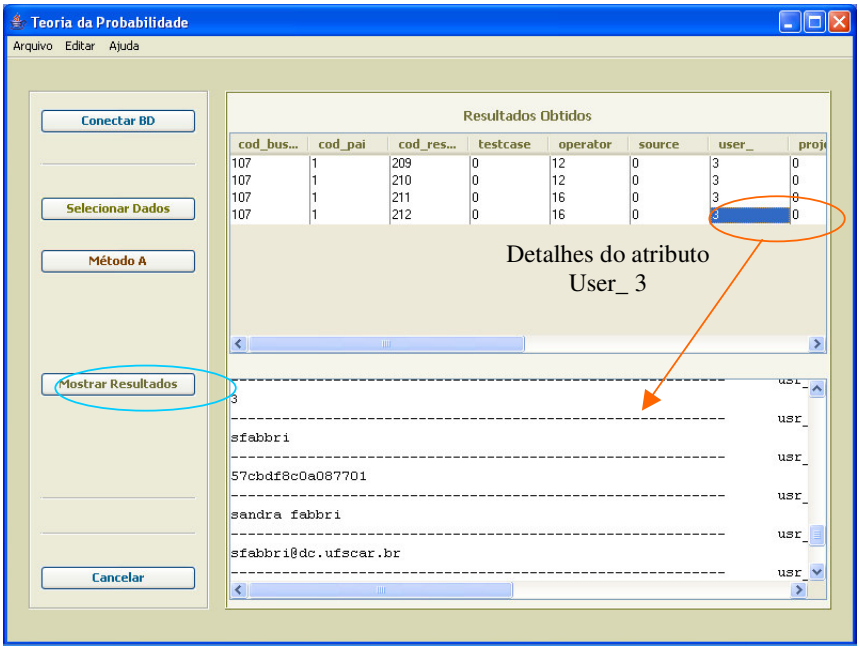


Figura 3.16 – Tela de resultados da probabilidade a priori.

- **Método B**

O Método B representa as probabilidades condicionais ou posteriores que são representadas por P(a|b), sendo a "a probabilidade de a, dado que tudo o que se sabe é b". A Tabela 2 ilustra a classificação de mutante e caso de teste:

Tabela 2 - Exemplo de resultados de busca

Mutante Caso Teste	Mutante Vivo	Mutante Morto	Total
TC1	16	30	46
TC2	20	10	30
Total	36	40	76

Tendo como exemplo estas informações armazenadas no *schema* PLAVIS, conforme exemplificado na Tabela 2, a ferramenta pode buscar a quantidade de mutantes vivos ou mortos em relação ao caso de teste. Por exemplo:

- Quando se deseja saber a probabilidade de se ter Mutante Vivo com o TC1. Utilizando a probabilidade condicional pode-se obter a seguinte informação:

P(MutanteVivo) = 36/76 = 0,47,

P(CasoTeste2) =20/76 = 0,27,

Probabilidade condicional P(CasoTeste2| MutanteVivo) = 20/76|36/76

= 0,27/0,47

= 0,57.

Chega-se a conclusão que a probabilidade de se ter mutantes vivos com o caso de teste 2 é de 57%.

Após ter escolhido o método e definidos quais dados deseja que seja realizado o cálculo, o sistema solicitará que seja informado qual tipo de resultado será mostrado. Caso a escolha seja pelo método A, tem-se apenas uma alternativa, que é apenas realizar os cálculos e informar os resultados clicando no botão “Mostrar Resultados”. Mas se a escolha for o método B o sistema solicitará dois tipos de resposta a Distribuição Total e a Tabela de Resultados.

Na opção Distribuição Total o sistema resolverá o seguinte problema, por exemplo:

Quando se têm vários casos de testes que age sobre um ou vários mutantes cria-se desta forma uma rede com n camadas como ilustrada na Figura 3.17.

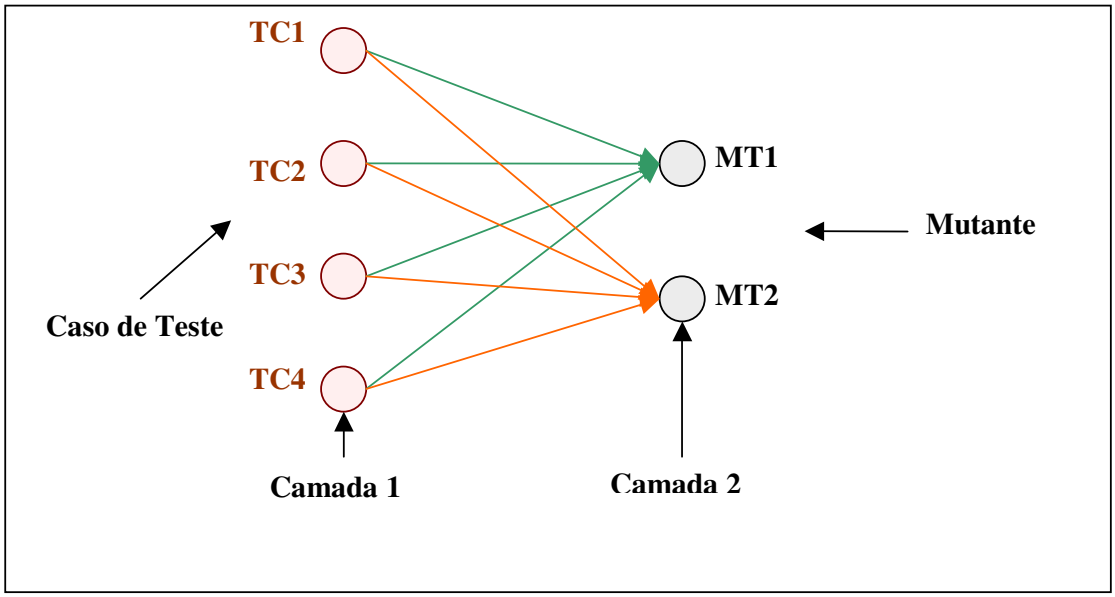


Figura 3.17 - Casos de testes que age sobre um ou vários mutante

Em cada nó contém uma probabilidade condicional que representa a ação do TC (Caso de Teste) em relação a um MT (Mutante), portanto, o sistema coleta os valores da camada *Um* junto com o valor do mutante, que possui resultado da equação de probabilidade incondicional, realizando o cálculo da representação da distribuição conjunta total.

Por exemplo:

P(TC1 ^ TC2 ^ TC3 ^ TC4 ^ MT1) =

*P(TC1|MT1) * P(TC2|MT1) * P(TC3|MT1) * P (TC4|MT1) * P(MT1) =*

*0,30 * 0,15 * 0,21 * 0,68 * 0,998 = 0,00064.*

P(*TC1|MT1*) - Probabilidade do Caso de Teste 1 deixar o Mutante Vivo 1;

P(*TC2|MT1*) - Probabilidade do Caso de Teste 2 deixar o Mutante Vivo 1;

P(*TC3|MT1*) - Probabilidade do Caso de Teste 3 deixar o Mutante Vivo 1;

P(*TC4|MT1*) - Probabilidade do Caso de Teste 4 deixar o Mutante Vivo 1;

P(*MT1*) - Probabilidade de Mutante Vivo 1;

Conclui-se que a probabilidade dos quatro casos de teste deixar o mutante *Um* ainda vivo depois da interação é de 0,064%.

Na opção Tabela de Resultados ocorre o mesmo cálculo, mas com todas as alternativas possíveis. Cada nó das camadas sucessoras forma uma tabela que ilustra as interações que ocorre na camada, informando as possíveis possibilidades de ocorrência das entradas da camada anterior.

Na Figura 3.18 é mostrado como ficam as interações dos casos de teste nos mutantes, gerando a tabela de probabilidade.

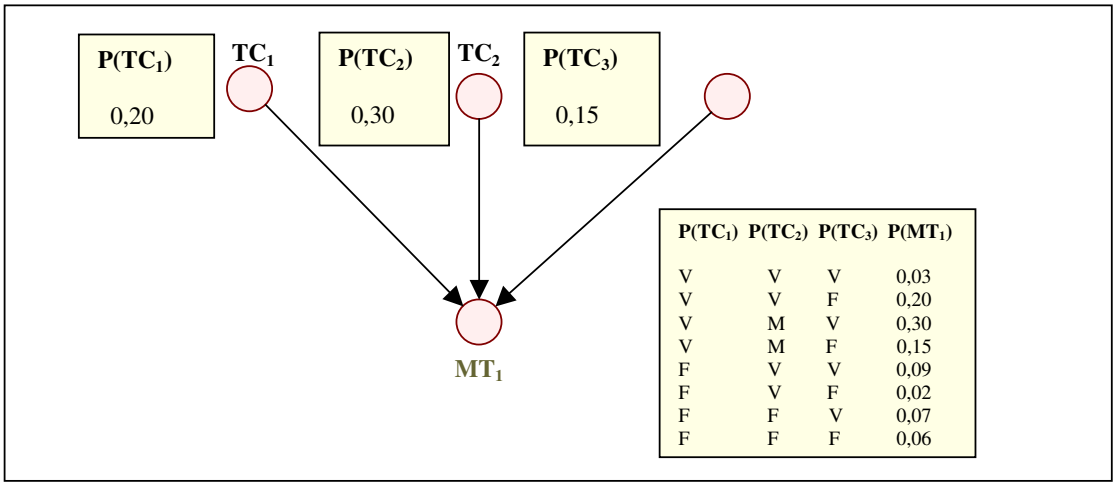


Figura 3.18 – Geração da Tabela de Probabilidade

Na tabela gerada para o MT1 (Mutante 1) dependendo da quantidade de nós que existe na camada anterior, a quantidade de possibilidades que podem gerar é da ordem de 2^n. Em que os cálculos são obtidos a partir da representação da distribuição conjunta total sendo calculado para cada linha da tabela.

Com a análise feita na tabela de probabilidade de cada nó chegou-se a conclusão que possui o mesmo conceito da Tabela Verdade. Por exemplo, quando temos três valores na camada superior que significa $2^3 = 8$ de resultado esperado, com isto tem-se $2^2 = 4$ na primeira coluna formando uma seqüência de verdadeiro e falso de

quatro em quatro e na segunda coluna tem se $2^1 = 2$ formando uma seqüência de dois em dois e na terceira e última coluna tem se $2^0 = 1$ formando uma seqüência de um em um.

Para realizar este cálculo foi desenvolvido um algoritmo que ao dividir o valor pelo número 2 e o resto contiver o valor 0 o algoritmo adquirirá o valor verdadeiro e, se a resposta contiver o valor 1 o algoritmo adquirirá o valor falso. Este resultado será utilizado para calcular cada linha da tabela (Ver APÊNDICE B – Cálculo Probabilístico).

3.7 Busca de melhor resultado

Para se buscar a melhor decisão, levando em consideração as preferências (utilidades) do usuário, foi usada a Teoria da Utilidade. Com seu conceito, anteriormente definido, busca-se, no início do processo oferecido pela ferramenta, o que o usuário deseja obter na seleção dos dados, assim como ilustrada na Seção 2.2.

Baseado nas preferências do decisor, calcula-se, portanto, as probabilidades desejadas com a utilização do Teorema de Bayes, conforme explicado na Seção 2.2, sendo possível, também a escolha do método que o usuário deseja utilizar para este cálculo.

Essas probabilidades são usadas no cálculo da utilidade, conforme citado por Jansen (2004, p. 2257), porém com uma forma que se diferencia do conceito do autor, que é a não utilização de árvore de decisão. Mesmo tendo a vantagem de ser uma representação muito clara e útil, o motivo de tal escolha deve-se ao fato de que há uma desvantagem considerável com o uso da mesma, já que esta ferramenta deve poder suportar elevadas bases de dados. Tal desvantagem é gerada quando o número de

variáveis for elevado, pois a árvore de decisão obtida pode tornar-se confusa, segundo Ladeira, Coelho e Viccari (1998, p. 41).

Baseado em Jansen (2004, p. 2257), o processo foi desenvolvido da seguinte forma:

A busca pelas preferências do decisor inicia-se quando o usuário clicar na opção da tela “Selecionar Dados”.

Estruturou-se o problema em uma Matriz de Decisão, que é uma tabela onde foram lançados nas colunas os elementos do problema como os dados escolhidos para fazerem parte do processo e as probabilidades;

Buscou-se o valor máximo da utilidade esperada, sendo que este varia de 0 a 1, atribuindo-se o valor 0 ao pior valor e o valor 1 ao melhor valor, tendo também os valores intermediários.

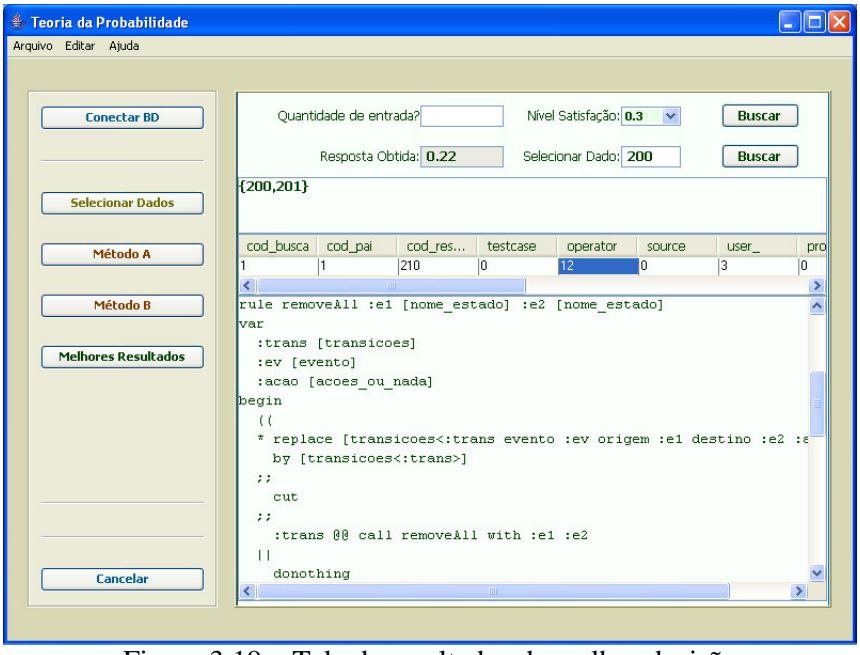


Figura 3.19 – Tela de resultados da melhor decisão

Em seguida, é efetuada a busca dos objetivos em cada uma das alternativas possíveis.

Assim, permite-se escolher a alternativa que mais se adequar às preferências do decisor, que é a que apresentar maior valor de utilidade.

A melhor decisão calculada para este decisor é mostrada no campo “Resposta Obtida” da Figura 3.19.

A ferramenta permite ao usuário, caso deseje, deixar suas preferências mais focadas, escolhendo, como por exemplo, a quantidade de entradas que deseja ter em sua busca.

Também é possível verificar as informações armazenadas para um determinado dado, estas serão mostradas na segunda tabela da tela. Se esse dado não pertencer à Base de Dados atual, será mostrada uma mensagem ao decisor sobre isso, conforme ilustrado na Figura 3.20.

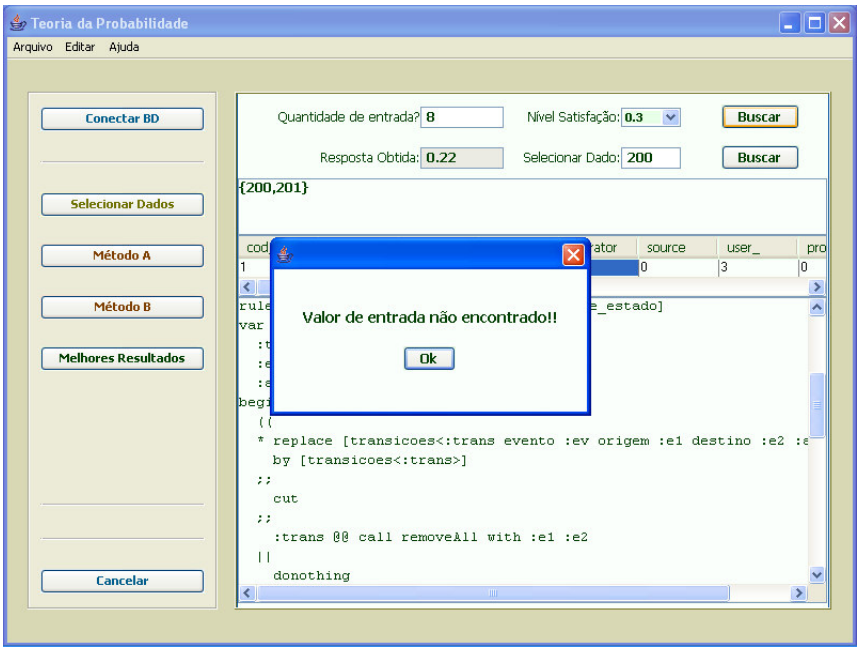


Figura 3.20 – Tela de mensagem de erro

3.8 Considerações Finais

No Capítulo 3 foram apresentados as funcionalidades da ferramenta DaMiTeCa, o modo de interação do usuário com a mesma e como foi o seu desenvolvimento, mostrando também as opções possíveis que o usuário pode escolher para a obtenção do resultado.

No Capítulo 4 são mostrados os resultados obtidos com a elaboração deste trabalho.

4 RESULTADOS OBTIDOS

Com a escolha do método A, o usuário tem como resposta as probabilidades a priori. Os resultados obtidos são simples em relação a método B. É utilizado para descobrir a probabilidade que ocorre entre as intersecções da tabela.

Um exemplo de resultado obtido utilizando o método A é quando o programa busca na tabela de execução de casos de teste (mutantexecutionstatus) um valor x, significando que existem 50 mutantes inseridos nesta tabela e o usuário deseja saber a probabilidade do mt_1 estar vivo. Uma outra consulta na base de dados obteve 20 mt_1 (mutante 1) vivos, retornando uma probabilidade de 20/50 = 0,4 do mt_1 estar vivo.

Também se pode citar o exemplo em que o usuário seleciona três mutantes como sendo mt_1, mt_2, mt_3 e deseja saber a quantidade de vezes que cada mutante ficou vivo. Assim, a ferramenta busca a quantidade de vezes que os mutantes permaneceram vivos como mt1v = 10, mt2v = 20, mt_3v = 15 e a quantidade de mutantes que foram executados como sendo mt_1 = 30, mt_2 = 30 e mt_3 = 30 vezes, tendo, portanto, a probabilidade de mt_1v = 30%, mt_2v = 66%, mt_3v = 50% deles estarem vivo

Pode-se, então, concluir que o mt_2 teve uma quantidade de casos de testes que trouxe um resultado menos satisfatório que os casos de testes realizados nos mt_1 e mt_3.

Com a escolha do método B que é responsável por construir tabelas probabilísticas que retorne resultados das interações das tabelas, demonstrou-se que é possível executar uma quantidade menor de interação para obter o resultado esperado. E também mostrou que dependendo dos dados selecionados pode-se obter uma melhor resposta, para um determinado caso. Na Figura 4.1, estão várias ilustrações da obtenção dos resultados.

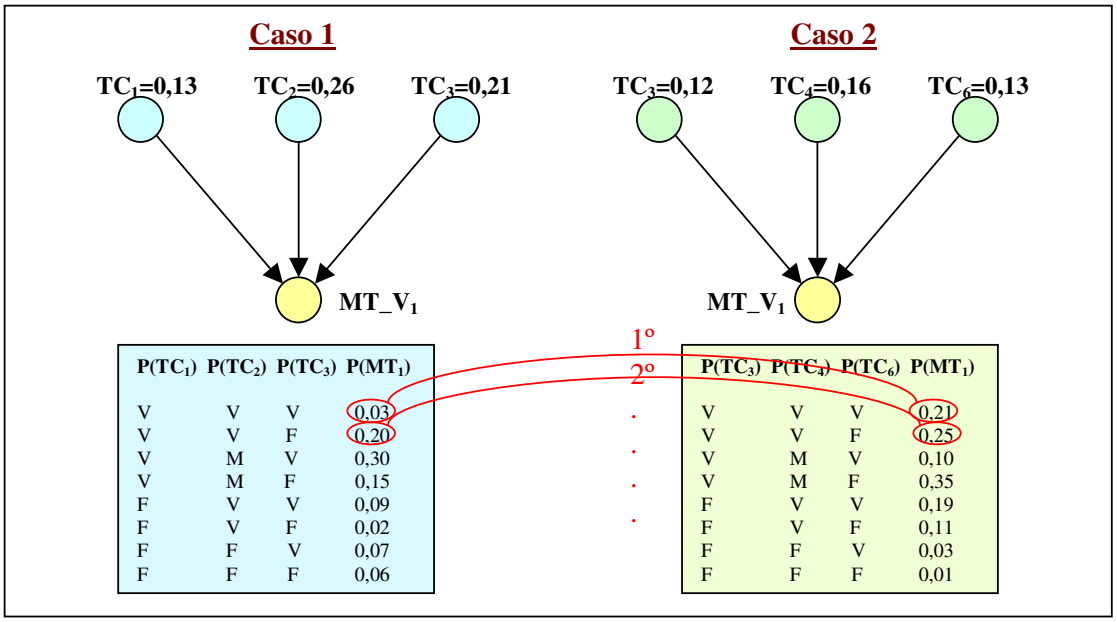


Figura 4.1 – Ilustra os resultados 1

Na Figura 4.1 está demonstrado que dependendo dos valores dos casos de teste que ocorrer de entrada pode trazer diferentes resultados no Mutante. Neste exemplo foram ilustrados dois casos com a mesma quantidade de mutantes de entrada, mas que obteve resultados diferentes. Na 1ª linha o primeiro caso teve uma resposta de 0,03, enquanto no segundo caso teve uma resposta de 0,21, portanto, o sistema demonstrou que os casos de teste TC_3, TC_4 e TC_6 quando agem juntos sobre o mutantes 1 tem uma maior chance de deixar o mutantes vivo do que em relação ao mutante TC_1, TC_2 e TC_3.

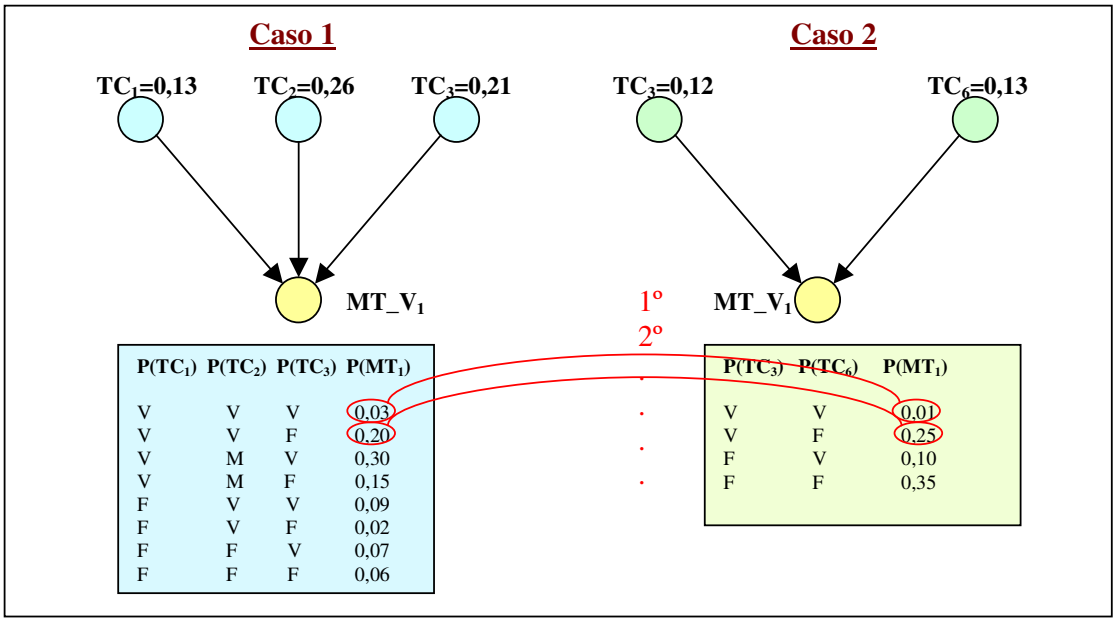


Figura 4.2 – Ilustra os resultados 2

Na Figura 4.2 é demonstrado o caso 1 com uma quantidade de entrada inferior ao do caso dois. Pode-se chegar a conclusão de que nem sempre uma quantidade maior de casos de testes tem uma maior chance de deixar o mutante vivo. Como mostrado na Figura 4.2 acima o primeiro caso tem três casos de teste e obteve uma resposta de 0,03% de chance de o mutante estar vivo, enquanto que no segundo caso tem dois casos de teste e obteve uma resposta de 0,01% de chance de o mutante estar vivo, informando que os casos de teste TC3 e TC6 deixaram mutantes mais vivos do que o caso um.

CONCLUSÃO

Com a definição de como extrair informações da base de dados utilizando a técnica de mineração de dados foi possível um melhor entendimento dos conceitos para a elaboração da ferramenta e pôde-se concluir que esta é de extrema importância, pois permite a escolha de características desejadas, isto é, um conjunto de casos de teste, e não mais um volume elevado de informações como se tem na maioria das ferramentas de apoio de teste de *software*.

A ferramenta, conforme já mencionado, armazena todos os casos de testes gerados nas etapas de testes e também os resultados obtidos em cada execução, formando o conjunto de casos de teste escolhido por meio de preferências do decisor, assim, é possível representar e raciocinar com preferências do decisor devido às funcionalidades da ferramenta desenvolvida com base na Teoria da Utilidade proposta. Isto foi unido com as funcionalidades da Teoria da Probabilidade, o que possibilitou a geração da Teoria da Decisão que auxilia, conforme o nome sugestivo, na tomada da melhor decisão possível dentro da escolha do usuário.

Embora não tenham sido encontrados trabalhos diretamente relacionados com a junção destas técnicas para atingir o mesmo objetivo desta monografia, não permitindo uma comparação mais detalhada de resultados, pôde-se observar que foi uma boa alternativa. Com a definição de mineração de dados, Teoria da Utilidade e Teoria da Probabilidade foi descoberto que juntando os conceitos de mineração de dados (que é um processo que está sendo muito requisitado atualmente em várias instituições para diferentes fins como citados no texto anteriormente), e a tecnologia de Redes Bayesianas (que faz medições para uma finalidade de casos), chegou-se a

conclusão que este é um dos melhores métodos analíticos disponível para a tomada de decisão.

Porém, mesmo com resultado satisfatório da ferramenta conforme citado acima, foi observado alguns pontos negativos. A primeira observação negativa foi quanto à utilização das redes bayesianas não indicada para este trabalho, por um elevado grau de processamento dos dados. A segunda observação foi que devido à escolha das preferências do decisor, o resultado pode ser influenciado pela possível variação dos pesos das utilidades.

Assim, para que este trabalho pudesse ser melhorado, foi proposta a eliminação destes pontos negativos, conforme citado nos trabalhos futuros.

Trabalhos Futuros

Como trabalho futuro foi percebida a necessidade da utilização de um outro modelo de redes bayesianas, pois a utilizada neste trabalho possui uma deficiência quando se tem vários valores de entrada, que gera uma tabela grande de possibilidades de resposta, aumentando, assim, o grau de processamento dos dados.

Portanto, para resolver este problema pode-se citar a construção de um novo modelo de rede bayesiana que são as chamadas “Redes Bayesianas Dinâmicas” que tem como finalidade resolver o problema das várias variáveis de entrada, gerando um número menor de possibilidades de entrada, como conseqüência, gera um número menor de saída.

Já na teoria da utilidade, pôde-se observar a relevância da análise da , como forma da influência da variação dos pesos no resultado da decisão, já que os mesmos afetam diretamente o resultado. Como uma das alternativas encontradas para isso, é a

análise econômica comparativa entre as duas melhores alternativas obtidas, conforme citado por Jansen (2004).

REFERÊNCIAS

AGRAWAL, R. & SRIKANT, R. **Fast algorithms for mining association rules in large databases**. In: Proceedings of the 20th International Conference on Very Large Databases, pg. 487–499, 1994.

ALVES, Ronnie; BELO, Orlando. **Mineração de dados em sistemas multidimensionais.** 10 f. Universidade do Minho, Braga, Portugal, 2003.

BACARDIT, J. and BUTZ, M. V. **Data mining in Learning Classifier Systems**: Comparing XCS with Assist. In: 7th International Workshop on Learning Classifier Systems - IWLCS, Seattle, USA, 2004.

BERRY, M. J. A. and LINOFF, G. **Data mining techniques**. USA: Wiley Computer Publishing, 1997.

CUSINATO, Rafael Tiecher. **Teoria da decisão sob incerteza e a hipótese da utilidade Esperada**: conceitos analíticos e paradoxos. 2003. 181 f. Dissertação (Pós-graduação em Economia da Faculdade) - Ciências Econômicas da UFRGS, Ciências Econômicas da UFRGS, Porto Alegre, 2003.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados.** 4. ed. Editora Pearson, São Paulo – SP, pg. 4, 646 e 647, 2005.

FAYYAD, U.; G. PIATETSKY-SHAPIRO and SMYTH, P. **The KDD Process for Extracting Useful Knowledge from Volumes of Data**. In: Communications of the ACM, vol. 39, pg. 27–34, 1996.

FRAWLEY, W.; MATHEUS, C. J. and PIATETSKY-SHAPIRO, G. C. M. **Knowledge discovery in databases**: An overview. Journal: AI Magazine, vol. 13, pg. 57-70, 1992.

GOMES, Luiz F. A. M; FREITAS JÚNIOR, Antonio A. **A importância do apoio multicritério à decisão na formação do administrador.** Revista ANGRAD, v.1, n.1. Rio de Janeiro, jul./set.2000.

GONZALEZ, A.; DOUGLAS, D. **The Engineering of Knowledge-Based Systems**. New Jersey: Prentice Hall, 1993.

GUGLIELMETTI, Fernando Ribeiro; MARINS, Fernando Augusto Silva; SALOMON, Valério Antônio Pamplona. **Comparação teórica entre métodos de auxílio à tomada de decisão por múltiplos critérios**. In: XXIII ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 2003, Ouro Preto. ENEGEP 2003. Ouro Preto: out. 2003. p. 6.

JANSEN, Leila Keiko Canegusuco; SHIMIZU, Tamio; JANSEN, José Ulisses. **Uma análise de investimentos considerando fatores intangíveis**. In: XXIV ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 2004, Florianópolis. ABEPRO. Florianópolis: nov. 2004. p. 8. p. 2256-2263.

KOCK, Jr.; N. F.; MCQUEEN, R. J. and CORNER, J. L. **The nature of data, information and knowledge exchanges in business processes**: Implications for process improvement and organizational learning. In: The Learning Organization, vol. 4(2), pg. 70–80, 1997.

LADEIRA, M., FLORES, C. D.; HOHER, C.; VICARI, R. M. **Tratamento Eficiente da Incerteza em Sistema de Apoio à Decisão**. 27º SEMISH da SBC. Curitiba, 2000.

LADEIRA, Marcelo; COELHO, Helder; VICCARI, Rosa Maria. **Uma Arquitetura Multiagente para Tomada de Decisão em Ambiente com Incerteza.** 1998. 66 f. Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil, 1998.

LIU, B. HSU, W. **Post-analysis of learned rules**. In: AAAI, vol. 1, pg. 828–834, 1996.

MEYER, P. L. **Probabilidade Aplicações e Estatísticas**. ed. 2, pg. 42-60, 1965.

OHMUKAI, Ikki; HAMASAKI, Masahiro; TAKEDA, Hideaki. **A Proposal of Community-based Folksonomy with RDF Metadata**. In: Terceira Conferência de Web Semântica Internacional (ISWC2004), 2004.

PAZZANI, M.; MANI, S.; and SHANKLE, W. **Comprehensible knowledge discovery in databases**. In: Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society, Lawrence Erlbaum, pg. 596–601, 1997.

POSTGRESQL GLOBAL DEVELOPMENT GROUP em SITE OFICIAL POSTGRESQL. **Official Documentation.** Disponível em: http://www.postgresql.org/ Acesso em 20 de out. de 2006.

REZENDE, S.O.; PUGLIESI, J. B.; MELANDA, E. A. e PAULA, M. F. **Mineração de Dados**. In: Solange Oliveira Rezende. (Org.). Sistemas Inteligentes - Fundamentos e Aplicações 1. ed. Barueri, SP, vol. 1, pg. 307-336, 2003.

RUSSELL, Stuart Jonathan; NORVIG, Peter. **Inteligência Artificial.** 2. ed. Editora Elsevier, Rio de Janeiro – RJ, pg. 447-624, 2004.

SILBERSCHATZ, A. and TUZHILIN, A. **On subjective measures of interestingness in knowledge discovery**. In: Knowledge Discovery and Data mining, vol. 1, pg. 275–281, 1995.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados.** 3. ed. Editora Makron Books, São Paulo – SP, pg. 1-4, 701-703, 706-714, 1999.

SILVA, W. T. e LADEIRA, M. **Mineração de dados em redes bayesianas**. In: Anais do XXII Congresso Brasileiro de Computação SBC, vol. 2, pg. 235–286, 2002.

SPITERI, Louise. **Controlled Vocabulary and Folksonomies.** Dalhouse University.

THEODORATOS, Dimitri; SELLIS, Timos. **Data Warehouse Configuration**. Universidade técnica nacional de Atenas.Atenas, Grécia.

WERNKE, Rodney; BORNIA, Antonio Cezar. **A Contabilidade Gerencial e os Métodos Multicriteriais**. Revista Contabilidade & Finanças FIPECAFI - FEA – USP. São Paulo, FIPECAFI, v.14, n. 25, pg 60 – 71, jan./abr. 2001.

ZHOU, Z. H. **Three perspectives of Data mining**. Journal: Artificial Intelligence, vol. 143(1), pg. 139–146, 2003.

APÊNDICES

APÊNDICE A – Intersecções dos dados

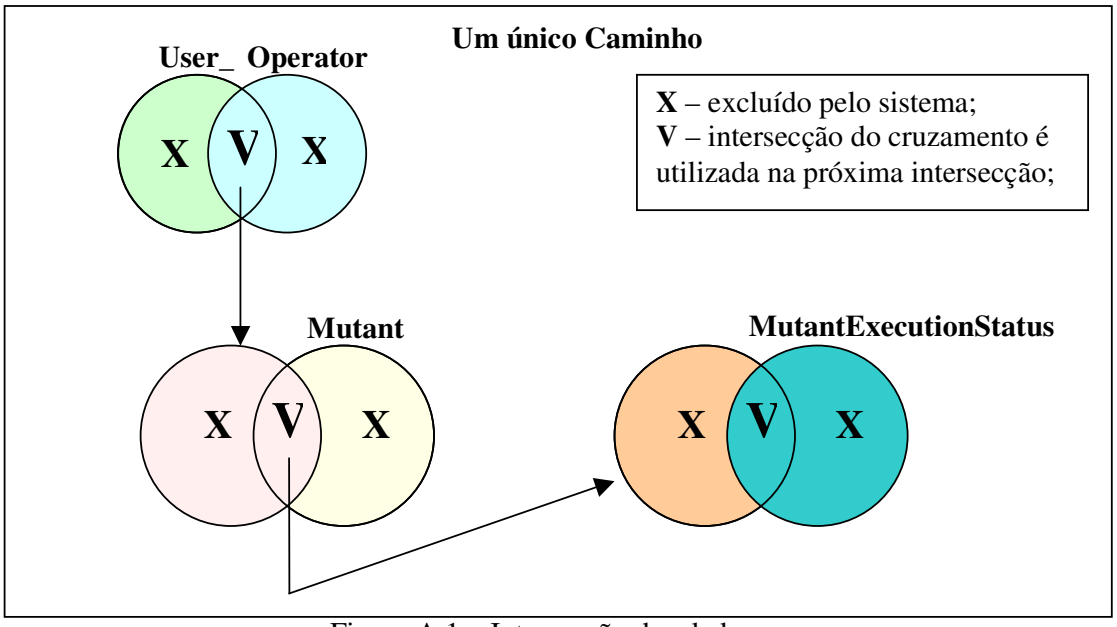


Figura A.1 – Intersecção dos dados

Na Figura A.1 é ilustrado um exemplo de intersecção das tabelas que foram selecionadas pelo usuário.

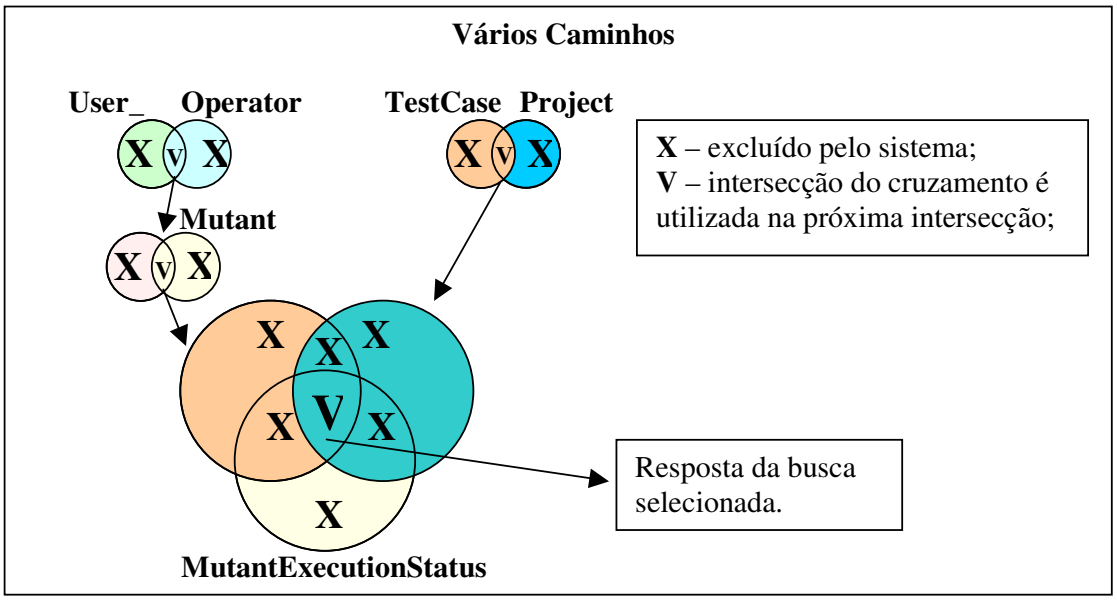


Figura A.2 – Intersecção dos Dados com Vários Caminhos

APÊNDICE B – Cálculo Probabilístico

O método ilustrado é realizado recursivamente em cada camada.

```
1 private void resultadoMetodo_TPC (String cod_busca, int maxpai)
2     {DM_datawarehouse dw1 = new DM_datawarehouse(
3                         Integer.valueOf(cod_busca).intValue(),
4                          Integer.toString(maxpai));
5     int qtidade_resultado = dw1.getcountCod_resultado();
6     cabecalho(cod_busca,maxpai,qtidade_resultado);
7     linhaRB = new Object[jTable1Model.getColumnCount()];
8     Double pow = Math.pow(2,qtidade_resultado); //exponenciação 2^n;
9     for(int y=1; y<=pow;y++)
10          {boolean continuaPrimeiro;
11          int nPrimeiro = 1;//int
12          String queryPrimeiro= dw1.SelectDB_11(cod_busca,maxpai);
13          continuaPrimeiro = 14
                  dw1.pegarDadosValorTabelaPrincipalDWBuscaResultado(
                  queryPrimeiro, 15nPrimeiro);
16          Double soma=1.0;
17          int calculo=y;
18          jTable1Model.addRow(linhaRB);
19          while (continuaPrimeiro)
20                {if(calculo%2==0)//resto igual a zero;
21                   {calculo=calculo/2;
22                    somay = somay+90;
23                    jTable1.setValueAt("Falso", n_linha,nPrimeiro-1);
24                }
25                else{calculo=calculo/2;//resto diferente de zero;
26                     soma =
27                     soma*Double.valueOf(DM_datawarehouse.
                         DH_resultado_calculoRB).doubleValu28e();
29                     somay = somay+90;
30                     jTable1.setValueAt("Verdadeiro",
                         n_linha,nPrimeiro-1);
31                }
32                nPrimeiro++;
33                continuaPrimeiro =
34                 dw1.pegarDadosValorTabelaPrincipalDWBuscaResultado(
                   queryPrimeiro, 35nPrimeiro);
36          }
37          jTable1.setValueAt(soma, n_linha,nPrimeiro-1);
38          n_linha++;
39          somay = somay+90;
40          somaz=somaz+10;
41     }
42    }
```

Figura B.1 - Código para realização do cálculo probabilístico

APÊNDICE C – Codificação

C.1 – Comandos do PostgreSQL

```
select relname as relname, relhaspkey
from pg_class, pg_tables
where relhaspkey = 'f'
        and pg_class.relname = pg_tables.tablename
        and SCHEMANAME = 'public';
```

O atributo relname representa o nome da tabela e o relhaspkey indica se a tabela possui ou não chave primária.

C.2 – Criação de tabela e funções dos atributos

```
CREATE TABLE datawarehouse.tabelaselecionada
      (cod_busca     numeric NOT NULL,
      cod_pai       numeric NOT NULL,
      cod_resultado numeric NOT NULL,
      ---------------------------------------
      Nome das tabelas que possui ligações com
  a tabelaselecionada.
       ---------------------------------------
       Nome dos atributos da tabelasselecionada.
       ---------------------------------------
       metodo        varchar(1),
      q_nometabela  numeric
      );
```

Os atributos têm o seguinte significado:

- *cod_busca*: armazena o código da realização da busca;
- *cod_pai*: armazena o código do caminho selecionado;
- *cod_resultado*: armazena os resultados obtidos em cada tabela selecionada;
- *metodo*: armazena o método que foi selecionado para a realização da busca;
- *q_nometabela*: armazena o resultado obtido na busca;

- *Nome dos atributos da tabelasselecionada*: Nome dos atributos: estes atributos armazenam os valores que foram selecionados para realizar buscas na base de dados PLAVIS.

- *Nome das tabelas que possui ligações com a tabelaselecionada*: estes atributos têm por finalidade armazenar os valores selecionados para fazer intersecção entre as tabelas.

C.3 – Criação da tabela no *Schema* datawarehouse.mutantexecutionstation

```
CREATE TABLE datawarehouse.mutantexecutionstation
( cod_busca numeric NOT NULL,
  cod_pai numeric NOT NULL,
  cod_resultado numeric NOT NULL,
  testcase numeric DEFAULT 0,
  "operator" numeric DEFAULT 0,
  source numeric DEFAULT 0,
  user_ numeric DEFAULT 0,
  project numeric DEFAULT 0,
  mutant numeric DEFAULT 0,
  mes_status varchar DEFAULT 0,
  mes_being_executed varchar DEFAULT 0,
  valor_final_calculado float8 DEFAULT 0,
  metodo varchar(1),
  q_mutantexecutionstatus numeric,
  CONSTRAINT mutantexecutionstation PRIMARY KEY (cod_busca, cod_pai, cod_resultado)
)
WITHOUT OIDS;
ALTER TABLE datawarehouse. mutantexecutionstation  OWNER TO postgres;
```

Tabelas que possui ligações

Atributos da tabela **mutantexecutionstation**

Figura C.1 - Código para realização do cálculo probabilístico

C.4 – Atributos e suas respectivas funções

Tabela Mutant:

- *mut_code:* código do mutante;

- *mut_diff:* guarda as informações da diferença entre o arquivo fonte e o mutante;

- *mut_status:* guarda a situação do mutante;

Tabela TestCase:

- *Tc_code:* código caso de teste;
- *Tc_imput:* entrada do caso de teste;
- *Tc_ouput:* saída do caso de teste;

Tabela MutantExecutionStatus:

- *Mes_status:* armazena a situação do mutante que pode ser mes1NotExec quando o mutante não for executado, mes3Dead quando o mutante está vivo, mes4Dead quando o mutante está morto e mes2Executing quando o mutante está sendo executado.
- *Mes_being_executed:* armazena S quando o sistema começou a ser executado e N quando não for executado.

C.5 – Retorna o código da tabela

```
select relfilenode as codtabela
from pg_class
where relname = 'nometabela';
```

C.6 – Retorna os atributos que não fazem parte de nenhuma outra tabela

```
select attname as atribnome
    from pg_attribute
    where attrelid = (codtabela)
    AND attstattarget = -1;
```

O atributo attname informa os nomes dos atributos com auxílio dos comandos attstattarget = -1 que informa somente os atributos que não são nem primário nem secundário.