



CENTRO UNIVERSITÁRIO UNIVATES
CURSO DE ENGENHARIA DA COMPUTAÇÃO

**ANÁLISE DA COMPRA DE PRODUTOS EM ESTABELECIMENTOS
COMERCIAIS UTILIZANDO A TÉCNICA DE MINERAÇÃO DE DADOS
“ANÁLISE DE AFINIDADES COM O ALGORITMO A PRIORI”**

Hélio Henrique Rodrigues Guimarães

Lajeado, dezembro de 2007



CENTRO UNIVERSITÁRIO UNIVATES
CURSO DE ENGENHARIA DA COMPUTAÇÃO

**ANÁLISE DA COMPRA DE PRODUTOS EM ESTABELECIMENTOS
COMERCIAIS UTILIZANDO A TÉCNICA DE MINERAÇÃO DE DADOS
“ANÁLISE DE AFINIDADES COM O ALGORITMO A PRIORI”**

Hélio Henrique Rodrigues Guimarães

Monografia apresentada na disciplina de Trabalho de Conclusão II, do Curso de Engenharia da Computação, como exigência parcial para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Alexandre Stürmer Wolf

Lajeado, dezembro de 2007

AGRADECIMENTOS

Agradeço a Deus, por me dar equilíbrio nas horas difíceis para a realização deste trabalho e por me impulsionar a seguir em frente sempre, independente dos obstáculos.

Aos meus queridos pais, Juarez e Vera, pelo incondicional amor, incentivo, e por serem sempre grandes exemplos de vida. Agradeço por entenderem os momentos de ausência, que não foram poucos, ao longo desta caminhada.

A minha namorada Jaqueline, pela sua paciência, carinho, e compreensão, principalmente neste último ano, durante o período de elaboração deste estudo.

Ao colega e amigo Nasair Júnior da Silva, pela indicação e por não medir esforços para a liberação da base de dados pelo estabelecimento estudado, o qual foi fundamental para elaboração deste trabalho.

A todos os sócios e proprietários do estabelecimento estudado, pelo excelente atendimento, dedicação e comprometimento com o autor, fornecendo todas as informações necessárias para a elaboração deste trabalho.

Ao meu orientador, professor Ms. Alexandre Stürmer Wolf, pelo acompanhamento, incentivo e grande contribuição teórica e prática, fundamentais para a realização deste estudo.

Aos professores da UNIVATES, pelos ensinamentos e experiências compartilhados.

A todos os amigos e colegas de curso, pelo empenho, ajuda e companheirismo, principalmente nos momentos extra-classe.

A todos que compartilharam e contribuíram de alguma forma a este estudo, muito obrigado!





“Você pode sonhar, criar e construir a idéia mais maravilhosa do mundo, mas são necessárias pessoas para fazer sonho virar realidade”.

Walt Disney

RESUMO

Esta monografia aborda a parte teórica da área de “descoberta de conhecimento em base de dados”, suas características gerais e a área de Mineração de Dados. Analisa “o que é um problema de Mineração de Dados” e como são propostas e desenvolvidas soluções nesta área computacional. Aplica metodologias e algoritmos específicos de Mineração de Dados, focando a identificação de quais produtos são mais vendidos em conjunto na organização estudada.

Palavras-chaves: Mineração de Dados. Descoberta de conhecimento em base de dados. Algoritmo *a priori*.

ABSTRACT

This monograph addresses the theoretical part of the area "knowledge discovery in databases," its general characteristics and the area of Data Mining. It analyzes "what is a Data Mining problem" and how are proposed and developed computational solutions in this area. It applies methodologies and algorithms specific to Data Mining, focusing on the identification of which products are most sold together in the studied organization.

Keywords: *Knowledge Discovery in Databases. Data Mining. A priori algorithm.*

LISTA DE ABREVIATURAS

- CSV Comma-separated values
- KDD *Knowledge Discovery in Databases*

LISTA DE FIGURAS

FIGURA 01 – Diferenças entre “descoberta de conhecimento em base de dados” e Mineração de Dados.....	18
FIGURA 02 – Processo de “descoberta de conhecimento em base de dados”	19
FIGURA 03 – Modelo de uma árvore de decisão.....	27
FIGURA 04 – Exemplo de uma rede neural.....	29
FIGURA 05 – Análise do carrinho de compras para entender comportamento de compra de um indivíduo.....	30
FIGURA 06 – Processo não linear de Mineração de Dados	34
FIGURA 07 – Formato de dados transacionais.....	38
FIGURA 08 – Representação tabular dos dados	39
FIGURA 09 - Estrutura e desenvolvimento da ferramenta	51

LISTA DE TABELAS

TABELA 01 – Lista de compras	42
TABELA 02 - Lista de 1- <i>itemsets</i>	42
TABELA 03 - Lista de 2- <i>itemsets</i>	42
TABELA 04 - Lista de 3- <i>itemsets</i>	43
TABELA 05 - Lista de 4- <i>itemsets</i>	43
TABELA 06 - Exemplo de estrutura de arquivo CSV para registro de compras.....	47

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 Objetivo	14
1.2 Objetivos específicos	14
1.3 Delimitação do estudo	15
1.4 Justificativa.....	15
2 REFERENCIAL TEÓRICO.....	17
2.1 Descoberta de conhecimento em base de dados	17
2.1.1 Seleção dos dados.....	19
2.1.2 Pré-processamento	20
2.1.3 Transformação	20
2.1.4 Mineração de dados.....	20
2.1.5 Interpretação e avaliação dos resultados	21
2.2 Mineração de dados.....	21
2.3 Metodologia da mineração de dados	22
2.3.1 Descoberta não supervisionada de relações.....	22
2.3.2 Teste de hipóteses	22
2.3.3 Modelagem matemática dos dados.....	23
2.4 Técnicas de Mineração de Dados.....	23
2.4.1 Classificação	23

2.4.2 Estimativa	24
2.4.3 Previsão	24
2.4.4 Análise de afinidade	25
2.4.5 Análise de agrupamentos.....	25
2.4.6 Sumarização	26
2.5 Algoritmos de Mineração de Dados	26
2.5.1 Árvores de decisão.....	26
2.5.2 Redes neurais	27
2.5.3 Análise de agrupamento.....	29
2.5.4 Regras de associação.....	29
2.5.5 Classificação Bayesiana.....	31
2.5.6 Algoritmos genéticos	31
2.5.7 Algoritmo <i>a priori</i>	31
2.5.8 Algoritmo <i>GRI</i>	32
3 METODOLOGIA	33
3.1 Identificação do problema de negócio.....	34
3.2 Seleção de dados apropriados.....	35
3.3 Conhecer os dados	35
3.4 Criação do modelo	35
3.5 Resolver problemas dos dados.....	36
3.6 Transformação dos dados para construção de informações para mineração dos dados	36
3.7 Construção do modelo	37
3.8 Avaliação do modelo	37
3.9 Interpretando o modelo	37
3.10 Avaliação dos resultados.....	38
3.11 Aplicação do algoritmo <i>a priori</i> da técnica de análise de afinidades da Mineração de Dados	38
3.11.1 Representação de dados para regras de associação	38
3.11.2 Características dos algoritmos de regras de afinidades.....	39
3.11.3 Algoritmo <i>a priori</i>	41

4 CARACTERIZAÇÃO DA EMPRESA.....	45
5 DESENVOLVIMENTO.....	46
5.1 Análise dos dados da empresa	46
5.1.1 Registro de compras	47
5.1.2 Registro de produtos	48
5.2 Pré-processamento dos dados.....	50
5.3 Estrutura de desenvolvimento	51
5.4 Ambiente de desenvolvimento	53
6 ANÁLISE DOS RESULTADOS	54
6.1 Base de dados	54
6.2 Mineração de Dados no estabelecimento	55
6.3 Resultados no estabelecimento	55
7 CONSIDERAÇÕES FINAIS.....	59
8 REFERÊNCIAS BIBLIOGRÁFICAS.....	61

1 INTRODUÇÃO

Nos dias de hoje, cada vez mais as organizações utilizam sistemas informatizados para controle de suas vendas, clientes, etc. Com o passar do tempo, a quantidade de dados existentes nos sistemas vai aumentando, tornando mais difícil a extração de conhecimento “útil”, fundamental para auxiliar no gerenciamento da organização.

Para tanto, tradicionalmente, as organizações optam por utilizarem técnicas da estatística tradicional, porém estas técnicas tornam-se inviáveis à medida que se tem, aumento no tamanho da base de dados a ser analisada. Outra característica da estatística tradicional é que primeiramente precisa-se criar uma hipótese, e posteriormente testar se é aceita ou refutada pelo conjunto de dados. Com isso, há grande intervenção humana no processo, dificultando o acesso rápido aos resultados, ocasionando muitas vezes defasagem no resultado, devido ao tempo de obtê-lo. Esse fato ocorre por a base de dados ser alimentada paralelamente, ou seja, quando se obtêm os resultados, já haverá nova base de dados composta pelos registros anteriores (onde foi feita a primeira análise) mais os registros que foram inseridos no período.

Para agilizar este processo e resolver os problemas da estatística tradicional, há a técnica das ciências computacionais denominada “descoberta de conhecimento em base de dados”, da área ligada à inteligência computacional e à estatística. Esta

técnica consiste em analisar os dados tentando encontrar padrões que venham a auxiliar na tomada de decisões.

Essa descoberta de conhecimento é dividida em etapas, sendo uma delas a Mineração de Dados, que consiste em analisar os dados e extrair padrões ou características muitas vezes não visíveis pelo analista. Ainda, dentro da própria Mineração de Dados, existem sub áreas que possuem características específicas entre elas, como resultados obtidos, entradas requeridas, limitações da área, algoritmos utilizados, etc.

Este estudo propõe-se a gerar resultados que visem o auxílio na tomada de decisões nas organizações por meio da aplicação de um algoritmo *a priori*, que faz parte de um conjunto de técnicas para análise de afinidades, principalmente utilizada na área de Mineração de Dados.

1.1 Objetivo

Desenvolver uma ferramenta de Mineração de Dados, utilizando o algoritmo *a priori*, para auxiliar as organizações na tomada de decisões por meio do estudo do comportamento de seus clientes.

1.2 Objetivos específicos

- Analisar o problema de inferência sobre grandes conjuntos de dados;
- Estudar a área de “descoberta de conhecimento em base de dados”;
- Analisar as técnicas de Mineração de Dados;
- Aplicar/Desenvolver um algoritmo para auxiliar nas inferências sobre grandes conjuntos de dados com foco na descoberta de vendas cruzadas de produtos;
- Aplicar a técnica proposta em um estabelecimento comercial;

- Analisar os resultados obtidos.

1.3 Delimitação do estudo

Observando-se as técnicas utilizadas nos dias de hoje para análise de grandes quantidades de dados, com o objetivo de descobrir relações entre os mesmos, nota-se que a maioria dos processos já são feitos com o uso de computação e softwares específicos para a análise de dados. Contudo, muitos analistas não possuem boas ferramentas que lhes forneçam resultados corretos, de forma clara e gerencial, para auxiliar no processo de tomada de decisão.

Com este princípio, o presente estudo foca a análise/adaptação da técnica de Mineração de Dados denominada “análise de afinidades”, para descobrir quais produtos são mais vendidos em conjunto no estabelecimento em estudo. O software permitirá a “eliminação” de soluções idealmente consideradas não boas, que só seriam eliminadas pelos processos tradicionais após análise dos dados gerados via software, bem como a descoberta de um conhecimento não notável.

Além dos critérios para avaliação de uma solução, se ela é boa ou ruim, fornecidos pelo software, é adequado que a pessoa que fizer a análise final dos dados seja alguém já envolvido no processo e que possa aliar a sua experiência com os resultados computacionais obtidos, para ter uma solução final de melhor qualidade.

1.4 Justificativa

Atualmente, ter o acesso à informação de forma rápida e confiável é um diferencial em relação à concorrência. Para esta premissa ser válida, é preciso ter confiança nos dados a analisar e identificar quais darão suporte a uma decisão mais confiável. Ainda, necessita-se ter a informação de forma rápida.

Cabe salientar que o dado é um registro de uma pessoa em um sistema; o registro de uma compra; o registro de uma venda etc. Porém, a informação é a resposta a uma pergunta, com relação à base de dados na sua totalidade, que

geralmente não é obtida de forma tão rápida. Um exemplo de informação seria o caso de ter-se um sistema de vendas e querer saber “quais CDs são mais vendidos às pessoas do sexo masculino entre 25 e 30 anos”.

Analisando esse exemplo, utilizando alguns softwares estatísticos, ter-se-ia a informação. Porém, aqui encontra a diferença entre a estatística e a Mineração de Dados, pois nesta não se teria essa hipótese, tanto do consumo de CDs como deste consumo por sexo ou por faixa de idade. O próprio algoritmo de Mineração de Dados é que descobre estas hipóteses. A diferença básica entre a estatística e a Mineração de Dados é, portanto, que na estatística tem-se a hipótese já definida, e na Mineração de Dados quer-se descobrir a hipótese, bem como seus resultados.

Nos capítulos subseqüentes deste trabalho são tratadas as questões teóricas envolvendo a área escolhida para estudo, técnica a ser utilizada, algoritmo utilizado, etc. Aborda-se a metodologia a ser usada no desenvolvimento da aplicação prática, como foi feito o desenvolvimento da ferramenta e uma análise dos resultados obtidos. Também é vista a caracterização da empresa escolhida para aplicação deste trabalho e uma conclusão final.

No capítulo 2, é apresentado o embasamento teórico das técnicas e áreas escolhidas para a busca da solução ao problema proposto.

2 REFERENCIAL TEÓRICO

Este capítulo oferece referências para melhor entendimento sobre os principais termos deste trabalho, “descoberta de conhecimento em base de dados” e Mineração de Dados, bem como sobre características gerais, tipos de técnicas utilizadas e alguns exemplos de algoritmos, destacando suas particularidades para a área de Mineração de Dados.

2.1 Descoberta de conhecimento em base de dados

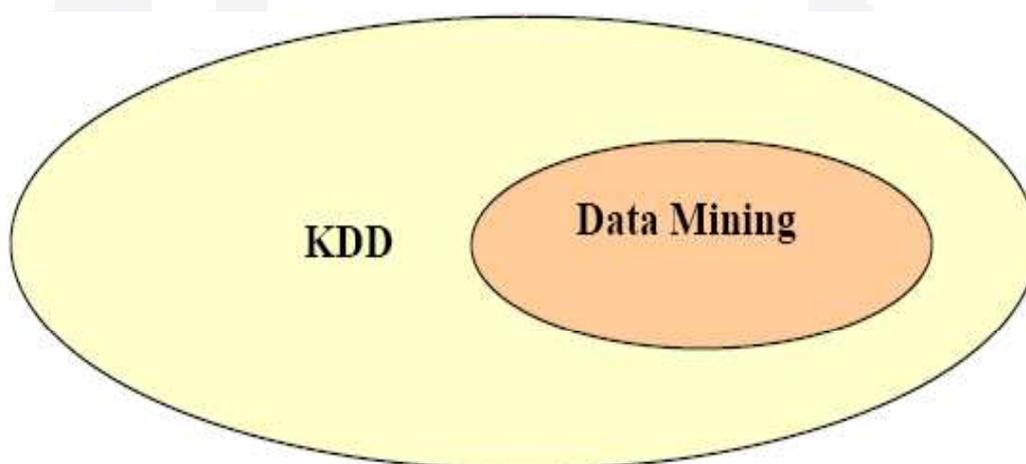
A “descoberta de conhecimento em base de dados” é uma área de estudo que consiste em analisar e descobrir conhecimento em grandes bases de dados existentes. Muitos dados são armazenados e possuem informações valiosas, como tendências e padrões que poderiam ser utilizados para melhorar as decisões de negócio. Com isso, tornou-se necessário o desenvolvimento de processos para análise automática de dados, como a Mineração de Dados (Rezende, 2005).

Rezende (2005, p.309) esclarece que “O processo de Extração de Conhecimento de Bases de Dados tem o objetivo de encontrar conhecimento a partir de um conjunto de dados para ser utilizado em um processo decisório”. Loshin (2003, p.205) também conceitua esse processo como “um meio de encontrar nova inteligência de uma coleção de dados”. Seu objetivo, segundo Lemos (2003, p. 15),

é desenvolver técnicas e ferramentas que buscam transformar dados armazenados em conhecimento.

Diferenciando “descoberta de conhecimento em base de dados” de Mineração de Dados, Lemos (2003, p.16) coloca que a primeira refere-se ao processo de descoberta de conhecimento útil em dados, enquanto a segunda compreende a aplicação de algoritmos para extrair estes dados. A Figura 01 apresenta uma comparação mostrando as diferenças entre a “descoberta de conhecimento em banco de dados”, abreviada na figura pela sigla KDD do significado em inglês *Knowledge Discovery in Databases*, e a Mineração de Dados, do significado em inglês *Data Mining*.

FIGURA 01 – Diferenças entre “descoberta de conhecimento em base de dados” e Mineração de Dados



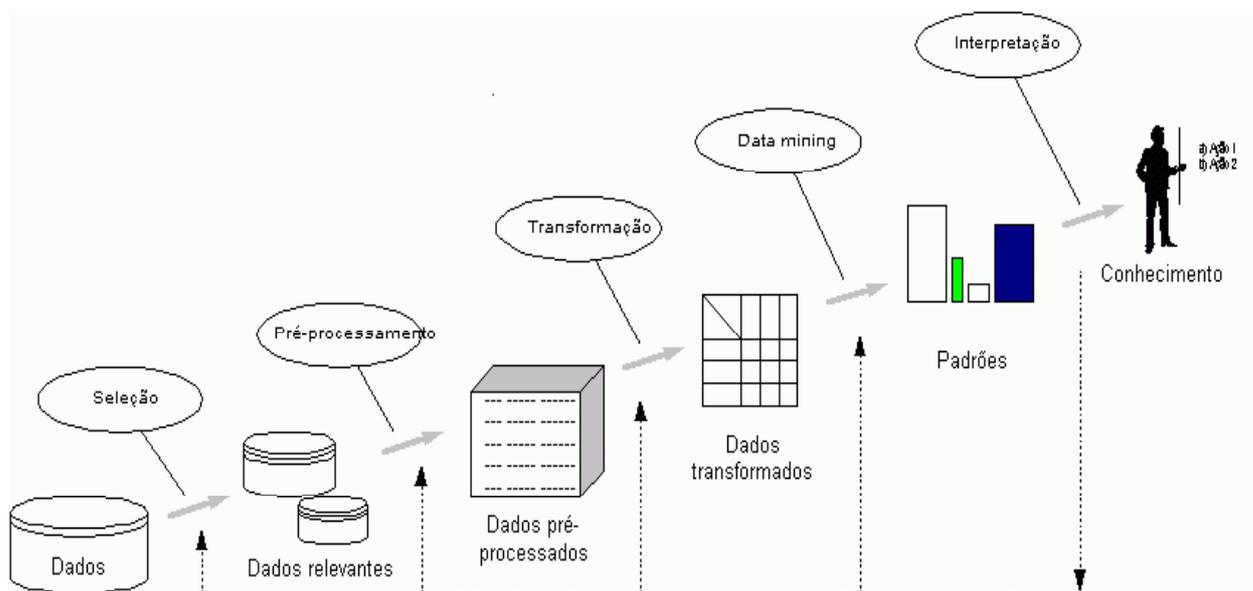
FONTE: Lemos, (2003, p.17).

O processo de “descoberta de conhecimento em base de dados” é dividido em cinco etapas, segundo Fayyad, Piatetsky-Shapiro e Smyth(1996) apud Lemos(2003) e Hand, Mannila, Smyth (2001):

- seleção dos dados;
- pré-processamento e limpeza dos dados;
- transformação dos dados;
- mineração de dados e
- interpretação e avaliação dos resultados.

A ordem dessas etapas pode ser vista na Figura 02:

FIGURA 02 – Processo de “descoberta de conhecimento em base de dados”



FONTE: Fayyad (1996) apud Lemos (2003, p.17).

Lemos (2003) ainda salienta que, apesar de existirem essas cinco etapas distintas, elas não são necessariamente executadas de forma linear, pois o resultado de uma pode obrigar ao retorno a alguma das etapas anteriores.

Nas próximas seções, são descritas as etapas do processo de “descoberta de conhecimento em base de dados”.

2.1.1 Seleção dos dados

Nesta etapa é feita a seleção ou segmentação dos dados de acordo com os objetivos de análise da base de dados (Cunico, 2005). Nela, segundo Carvalho, Sampaio, Mongiovi (1999), identificam-se as bases de dados que serão utilizadas e quais dados destas serão utilizados para a mineração.

Lemos (2003) salienta que muitas vezes nem todos os dados estão disponíveis em bases de dados adequadas, sendo necessário um trabalho de compatibilização dos dados.

2.1.2 Pré-processamento

No pré-processamento dos dados é decidido como será tratada a falta de dados em algum atributo, e também é realizada uma “limpeza” dos dados, sendo as informações julgadas desnecessárias removidas. Nesta etapa, ainda faz-se a configuração – re-configuração de dados para assegurar formatos consistentes (Cunico, 2005). Isto porque “[...] como os dados podem vir de várias bases, não necessariamente consistentes, impõe-se remover as inconsistências e integrá-los[...]” (Carvalho, Sampaio e Mongiovi, 1999, p. 4)

2.1.3 Transformação

Na terceira etapa, de transformação dos dados, faz-se a redução/diminuição da base de dados, aplicando-se técnicas/algoritmos. Nela deve-se transformar os dados para serem utilizados corretamente na etapa seguinte (Mineração de Dados), ou seja, é nesta etapa que se assegura que os dados estão prontos e corretos para executar a quarta etapa e auxiliar na inferência posterior dos mesmos (Cunico, 2005). Portanto, sua finalidade, conforme Carvalho, Sampaio, Mongiovi (1999), é compatibilizar os dados para servirem de entrada para o algoritmo escolhido de Mineração de Dados.

2.1.4 Mineração de dados

Na quarta etapa, é escolhida uma técnica/algoritmo para a utilização na aprendizagem do sistema de acordo com o objetivo a ser cumprido. É nela que realmente é feita a extração de padrões dos dados (Cunico, 2005).

Esta etapa será aprofundada na seção 2.2 deste trabalho.

2.1.5 Interpretação e avaliação dos resultados

Na última etapa, faz-se a interpretação dos resultados, devendo-se, ao constatar alguma incoerência, revisar e reaplicar as etapas anteriores à base de dados. Após estes procedimentos têm-se os padrões reconhecidos que darão suporte à tomada de decisão (Cunico, 2005).

2.2 Mineração de dados

Segundo Serra (2002), a Mineração de Dados é uma tecnologia que busca encontrar padrões, regularidades e tendências nos mais diversos bancos de dados, utilizando-se do poder computacional para encontrar informações até então desconhecidas dentro de um banco de dados. O autor destaca ainda que ela tem por objetivo extrair informações válidas, previamente desconhecidas e de máxima abrangência de grandes bases de dados.

Fayyad, Piatetsky-Shapiro e Smyth (1996a) apud Rezende (2003, p. 309-310), destacam que a Mineração de Dados “[...] é o processo de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis embutidos nos dados”.

Conforme Carvalho (2001, p. 6), “A Mineração de Dados consiste mais especificamente em descobrir relações entre produtos, classificar consumidores, prever vendas, localizar áreas geográficas potencialmente lucrativas para novas filiais, inferir necessidades, entre outras”. Hand, Mannila e Smyth (2004, p. 1) explica que: “Mineração de Dados é a análise dos (muitas vezes grandes) conjuntos de dados observacionais para encontrar relacionamentos insuspeitos e para descrever os dados de forma entendível e usável para o proprietário dos dados” .

Lemos (2003, p. 20) destaca que a Mineração de Dados difere de outras técnicas de análise de dados na maneira com que explora a inter-relação dos dados. Ou seja, as ferramentas de análise de dados tradicionais partem do princípio tradicional, pelo qual o analista constrói hipóteses sobre atributos da base de dados, e, por meio de teste, confirma ou refuta estas hipóteses. Já o processo de Mineração de Dados constrói estas hipóteses.

A autora explica que:

Existe muito conhecimento escondido na imensa quantidade de dados disponíveis nos bancos de dados das empresas. Com o Data Mining, pode-se transformar esses dados brutos em informação valiosa para auxiliar o processo decisório (Lemos, 2003, p. 25).

2.3 Metodologia da mineração de dados

Carvalho (2001, p. 17) salienta que, na escolha de uma determinada técnica/ algoritmo de Mineração de Dados a ser utilizada, deve-se levar em conta três metodologias para esta escolha, que são: “descoberta não supervisionada de relações”, “teste de hipóteses” e “modelagem matemática dos dados”.

2.3.1 Descoberta não supervisionada de relações

Esta técnica de Mineração de Dados é utilizada quando não se conhece o comportamento do fenômeno a ser estudado, deixando as técnicas/ algoritmos de Mineração de Dados procurarem na base de dados relações escondidas que não poderiam ser localizadas facilmente a olho nu. Por isso, essa técnica é descompromissada com qualquer relação predeterminada, sendo executada com a finalidade de “tentar descobrir” nova relação útil em um conjunto de dados (Carvalho, 2001).

Como a técnica gera muitas relações entre os dados, Carvalho (2001) esclareceu que, posteriormente a geração dos resultados, uma análise humana precisa ser feita para separar as relações interessantes das relações não interessantes.

2.3.2 Teste de hipóteses

Esta técnica é utilizada para a mineração/ análise de dados, quando já se possui algum conhecimento ou alguma idéia sobre qual “nova relação” está sendo buscada no conjunto de dados. Assim, cria-se uma hipótese em que a técnica/ algoritmo utilizada terá a função de confirmar ou refutar esta afirmação (Carvalho, 2001).

2.3.3 Modelagem matemática dos dados

Técnica que é utilizada quando se tem um grande conhecimento da área de estudo e das relações existentes entre o conjunto de dados escolhido. Com estes dados, conforme Carvalho (2001), pode-se partir do ponto que já se possui uma afirmação proposta, e analisa-se o comportamento numérico destes agrupamentos com relação a algum item que se julga interessante.

2.4 Técnicas de Mineração de Dados

Carvalho (2001) considera que existem muitas técnicas/algoritmos diferentes de Mineração de Dados, contudo, cinco técnicas contemplam todas as outras formas de apresentação e permitem uma visão mais global do assunto. Estas técnicas são: *classificação, estimativa, previsão, análise de afinidade e análise de agrupamentos*. Berry (2004) acrescenta a técnica de *sumarização*.

Berry (2004) também refere que essas técnicas são classificadas em dois grandes grupos: *diretas e indiretas*. As primeiras caracterizam-se por terem um atributo objetivo que se deseja mensurar. Nesta categoria enquadram-se as técnicas de *classificação, estimativa e previsão*. Já as segundas caracterizam-se por minerar e descobrir as relações com quaisquer atributos, sem ter um em específico. É o caso das técnicas de *análise de afinidade e análise de agrupamentos*. Já a técnica de *sumarização* pode enquadrar-se tanto em *direta* como *indireta*.

O autor destaca ainda que as técnicas diretas são chamadas também de *aprendizado supervisionado* e as indiretas de *aprendizado não supervisionado*.

2.4.1 Classificação

A técnica de classificação consiste, segundo Cunico (2005), em encontrar características comuns em determinado conjunto de dados, agrupando estas ocorrências em conjuntos de dados distintos. Por exemplo, pode-se dividir a população em categorias conforme a cor do cabelo de cada pessoa, em que estas

seriam classificadas e alocadas na categoria em que se enquadram. Neste exemplo, ter-se-iam as categorias: castanho, escuro, loiro, ruivo, etc.

Carvalho (2001, p. 21) explica que:

O homem, ao receber qualquer estímulo do meio ambiente, e preparando-se para uma resposta a ele, busca, por analogia, classificar esse estímulo em categorias de outros estímulos já recebidos no passado para os quais possui uma resposta pronta e imediata.

Dentre as várias técnicas existentes para a classificação, Cunico (2005) destaca as de: *Árvores de Decisão*, *Regressão Lógica*, *Redes Neurais* e *Algoritmos Genéticos*.

2.4.2 Estimativa

Segundo Carvalho (2001), estimar algum dado é determinar seu valor provável com base em dados do passado ou em outros dados semelhantes nos quais se tem conhecimento. Para Larose (2005), a estimativa é similar à classificação, porém na estimativa, o atributo alvo é numérico.

Entre as técnicas utilizadas, conforme Carvalho (2001), estão: Redes Neurais Artificiais, Estatística, Algoritmos Genéticos e *Simulated Annealing*. Berry (2004) ainda cita as árvores de decisão como algoritmos de solução para a técnica de estimativa.

2.4.3 Previsão

Previsão, para Carvalho (2001), é a técnica pela qual se deduz um “valor futuro” com base nos valores que se tem hoje. O autor ainda esclarece que a única forma de verificar se uma previsão foi bem feita ou não, é aguardando-se o acontecimento. E salienta: “Sem dúvida, a previsão é uma das tarefas mais difíceis não somente no *datamining*, mas também em nossa vida” (Carvalho, 2001, p.21).

No entender de Larose (2005), a previsão é similar à classificação e à estimativa, porém na previsão o resultado é para o futuro.

2.4.4 Análise de afinidade

Esta técnica estuda a probabilidade de eventos ocorrerem simultaneamente, ou seja, se um determinado evento ocorre juntamente com outro (Carvalho, 2001; Cunico, 2005). Como exemplo, podem-se extrair informações de quais produtos os consumidores compram em conjunto em um supermercado, possibilitando a realização de vendas dirigidas, ou até a elaboração de kits com um desconto de determinado grupo de produtos (Carvalho, 2001).

Larose (2005) aponta que o algoritmo *a priori* e o algoritmo *GRI* são utilizados na análise de afinidades.

O algoritmo *a priori*, será visto descritivamente na seção 2.5.7 e detalhadamente na seção 3.11.3. Já o algoritmo *GRI* será visto apenas descritivamente na seção 2.5.8 e pode ser visto com mais detalhes em consulta à bibliografia referenciada neste estudo. Isto porque o foco do estudo é no algoritmo *a priori*.

2.4.5 Análise de agrupamentos

Com esta técnica, classifica-se determinado conjunto de dados em classes e categorias, sem saber a quantidade de classes nem a quantidade de categorias existentes (Cunico, 2005).

No exemplo proposto por Carvalho (2001), tem-se a tarefa de agrupar sintomas de doenças, podendo-se gerar classes de sintomas que não representem nenhuma doença, uma vez que doenças diferentes podem apresentar o mesmo sintoma.

O algoritmo utilizado para esta técnica é chamado de *Análise de Cluster*. Este algoritmo pode ser aprofundado na bibliografia referenciada neste estudo.

2.4.6 Sumarização

Para Berry (2004), às vezes a mineração de dados tem apenas por objetivo descrever um acontecimento para facilitar o entendimento sobre pessoas, produtos, processos, etc, que produzem dados em um primeiro momento. Porém, estes dados não são de fácil entendimento somente com sua análise, e para suprir este problema, utiliza-se à descrição mais detalhada de determinado acontecimento para facilitar a interpretação dos mesmos. Esta técnica é chamada de *sumarização*.

2.5 Algoritmos de Mineração de Dados

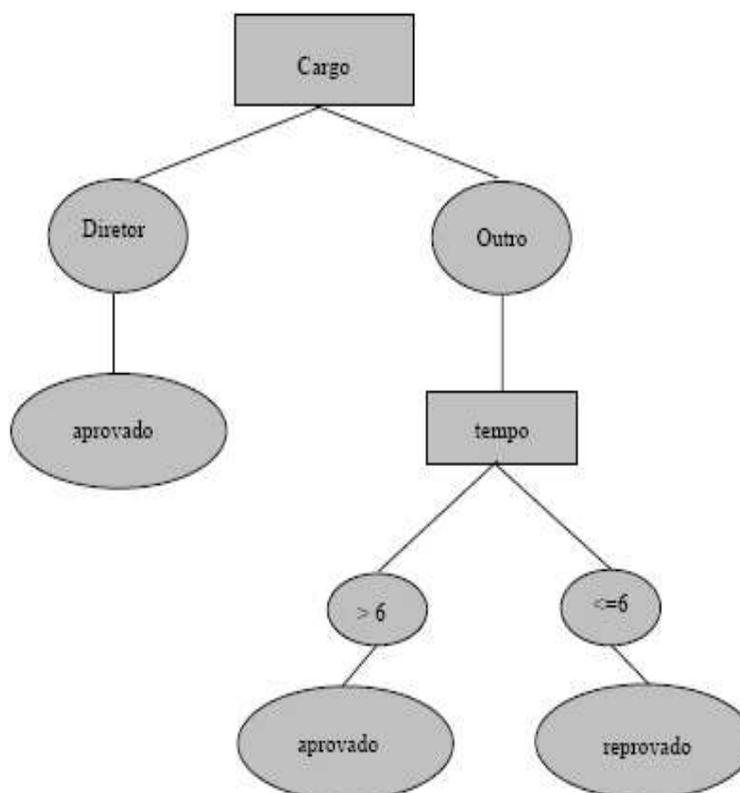
Na seqüência são citadas algumas características dos algoritmos de *árvores de decisão*, *redes neurais*, *análise de agrupamentos*, *regras de associação*, *classificação Bayesiana*, *algoritmos genéticos* e *algoritmo a priori*.

2.5.1 Árvores de decisão

Segundo Rud (2001), a meta das árvores de decisão é segmentar uma parte de dados para maximizar as diferenças das variáveis dependentes. Nelas, conforme Lima (2001, p. 60),

Os nós são rotulados com nomes de atributos, os arcos são rotulados com possíveis valores para este atributo e as folhas são rotuladas com diferentes classes. Objetos são classificados através de um caminho percorrendo a árvore - seguindo os arcos que contêm valores que correspondem a atributos no objeto.

Com as árvores de decisão, representa-se o resultado da Mineração de Dados na forma de árvores similares a um gráfico organizacional horizontal (organograma) (Cunico, 2001). A Figura 03 mostra um exemplo, no qual se aprovaria ou reprovava um pedido, conforme o cargo e o tempo de serviço na empresa para determinados cargos.

FIGURA 03 – Modelo de uma árvore de decisão

FONTE: Cunico (2005, p. 29).

Para Berry (2004), uma árvore de decisão é uma estrutura que pode ser utilizada para dividir grandes quantidades de dados em sucessivos grupos menores de registros, aplicando uma seqüência simples de regras de decisão.

Uma diferença ainda é que as arvores de decisão, em comparação com as outras técnicas, tem uma apresentação melhor dos resultados finais.

Em contra partida, força uma segmentação única: um dado só pode fazer parte de uma categoria de cada vez, e não a mais de uma.

2.5.2 Redes neurais

As redes neurais caracterizam-se por tentarem construir representações de modelos de conhecimento ou padrões detectados nos dados. Porém, estas

representações não são mostradas ao usuário, ficando ocultas dentro da rede neural (Lemos, 2003, p. 34).

Lemos (2003) e Cunico (2005) referem que, analisando a estrutura interna de uma rede neural, observa-se que ela consiste em um número de elementos interconectados, chamados neurônios, organizados em camadas, que aprendem pela modificação dos pesos de cada camada. Essas redes, conforme Lima (2001, p.59),

[...] são uma solução computacional que envolve o desenvolvimento de estruturas matemáticas com a habilidade de derivar médias de dados complicados ou imprecisos e podem ser utilizadas para extrair padrões e detectar tendências que são muito complexas para serem percebidas tanto por humanos quanto por outras técnicas computacionais.

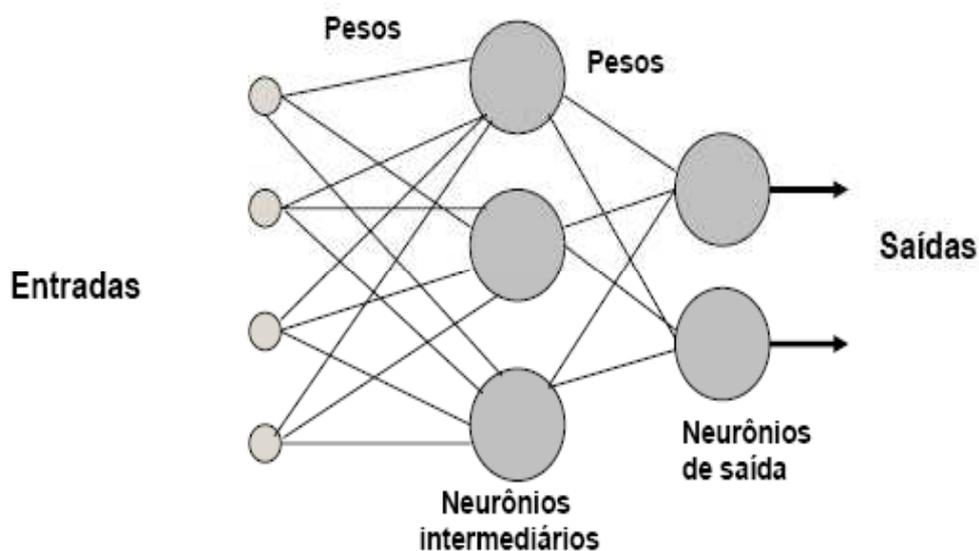
Lemos (2003, p.35) explica que:

As redes neurais artificiais utilizam um conjunto de elementos de processamento (ou nós) análogos aos neurônios no cérebro. Estes elementos de processamento são interconectados em uma rede que pode identificar padrões nos dados uma vez exposta aos mesmos, ou seja, a rede aprende através da experiência, tais como as pessoas.

Estruturalmente, uma rede neural consiste em um determinado número de elementos ligados (neurônios) organizados em camadas unidas por conexões. (Cunico, 2005)

Na Figura 04, demonstra-se um exemplo genérico de uma rede neural, composta pelas *entradas*, que seriam os dados da base de dados; os *pesos* atribuídos para avaliação destas entradas, os quais são calculados conforme a rede é treinada para melhorar seu aprendizado; os *neurônios intermediários*, que junto com os pesos não são vistos pelos usuários; os *neurônios de saída*, que possuem as saídas já relacionadas com seu peso; e, a *saída* que é vista pelo usuário (Lemos, 2003).

FIGURA 04 – Exemplo de uma rede neural



FONTE: Lemos (2003, p. 55).

2.5.3 Análise de agrupamento

Nesta análise, segundo Cunico (2005), o objetivo principal é identificar padrões em conjuntos de dados. Com isso, o algoritmo identifica comportamentos similares e divide este conjunto de dados em grupos. Assim a pessoa que analisará os dados pode verificar se estes resultados podem ser utilizados de forma estratégica ou não.

De acordo com Lemos (2003, p. 25),

Esta técnica agrupa informações homogêneas de grupos heterogêneos entre os demais e aponta o item que melhor representa cada grupo, permitindo desta forma que se consiga perceber a característica de cada grupo.

2.5.4 Regras de associação

As regras de associação, segundo Witten (2005), têm um funcionamento similar à classificação, porém, podem prever qualquer atributo do conjunto de

dados, não somente atributos classificáveis, dando liberdade também para a combinação destes atributos entre si.

O autor ressalta que, ao utilizar esta técnica, deve-se ter cuidado, pois muitas associações de regras podem ser derivadas do conjunto de dados, sendo interessante restringir esta derivação para alguns atributos (Witten, 2005).

Um bom exemplo de regras de associação, segundo Serra (2002), seria percorrer todos os registros de determinado ponto de venda qualquer, em uma loja, supermercado, etc, para descobrir que itens são vendidos juntos, de forma a redefinir a disposição de artigos na loja, ou criar campanhas publicitárias nas quais se pode explorar com maior eficácia essas associações.

Na visão de Cunico (2005, p.30), “[...] um bom algoritmo de extração de regras deve gerar regras que possuam suporte e confiança especificados pelo usuário e as regras podem ser compostas de um ou mais itens”.

Lemos (2003) diz que esta técnica separa os dados e agrupa-os para formar grupos homogêneos e diferentes entre si.

FIGURA 05 – Análise do carrinho de compras para entender comportamento de compra de um indivíduo



FONTE: BERRY (2004, p. 288)

2.5.5 Classificação Bayesiana

A técnica de classificação Bayesiana se enquadra nas técnicas de Mineração de Dados não supervisionadas. Nesta classificação, os dados não são divididos em grupos, mas cria-se uma função *probabilística* com base nas teorias de probabilidade de Bayes para cada grupo de dados (Cunico, 2005).

2.5.6 Algoritmos genéticos

Os algoritmos genéticos, segundo Cunico (2005), realizam a tarefa de classificação dos dados, utilizando como analogia as teorias de evolução genética e de seleção natural de Darwin. Esta idéia, conforme Berry (2004), parte do princípio de que cada indivíduo pode ser uma solução para um determinado problema proposto, sendo que estes indivíduos se reproduzem entre si.

Nos algoritmos genéticos, o objetivo é combinar os melhores indivíduos entre si, buscando um indivíduo melhor como solução para o problema. Esta busca ocorre até um critério de parada da execução do algoritmo. Este critério pode ser o tempo de execução, a avaliação da solução encontrada, etc (Berry, 2004).

2.5.7 Algoritmo *a priori*

Larose (2005) refere que o algoritmo *a priori* é um dos algoritmos que pertencem à técnica de análise de afinidades da Mineração de Dados. Como todos algoritmos pertencentes à técnica de análise de afinidades, este busca descobrir e quantificar a relação entre dois ou mais atributos de uma base de dados.

Ainda segundo Larose (2005), um dos problemas de alguns algoritmos de regra de associação é que o número de regras cresce exponencialmente com relação ao número de atributos. No caso, tendo-se k atributos, a relação de possibilidades de regras de associação são de k multiplicado por 2 elevado ao expoente k menos 1: $k \cdot 2^{k-1}$.

Por exemplo, tendo-se 100 diferentes atributos, tem-se como possibilidade de regras de associação:

$$100 \times 2^{100-1} = 100 \times 2^{99} \approx 6,4 \times 10^{31}$$

O algoritmo *a priori* para mineração de regras de associação utiliza algumas técnicas de análise da estrutura das regras, com o objetivo de reduzir o número de possibilidade de resultados (Larose, 2005).

2.5.8 Algoritmo *GRI*

Segundo Larose (2005), quando a estrutura das regras de associação possui um intervalo de dados numéricos, no qual os atributos não são caracterizáveis como no exemplo do algoritmo *a priori*, tem-se que utilizar outra técnica da análise de associações para resolver o problema. Esta técnica é a *geração de regras de indução – GRI*.

O algoritmo GRI parte do princípio de categorizar as variáveis numéricas. Por exemplo, categoriza-se a renda das pessoas, podendo-se criar as categorias: quem recebe menos de R\$500,00 iria se enquadrar na categoria *baixa renda*; quem recebe mais de R\$2000,00 se enquadraria na categoria *renda alta* e os demais na categoria *renda média* (Larose, 2005).

3 METODOLOGIA

Neste capítulo são descritas as etapas a serem seguidas para o processo de Mineração de Dados de uma base de dados qualquer. Pode-se, segundo Berry (2006), aplicar esta metodologia a todos os processos de Mineração de Dados, independentemente da técnica/algoritmo a ser utilizado.

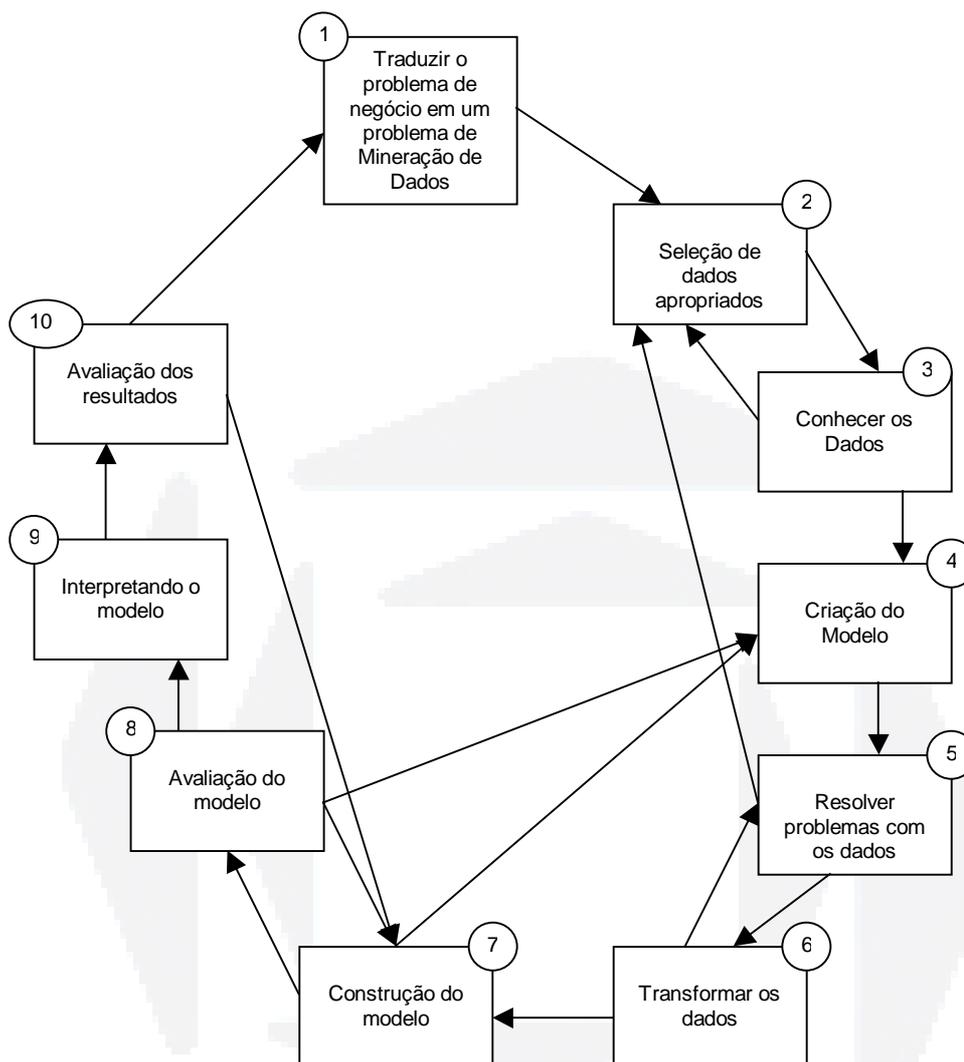
As etapas metodológicas de um processo de Mineração de Dados são divididas em dez itens, que são descritos nas seções 3.1 até 3.10.

Na seção 3.11 é descrito o algoritmo selecionado para a análise dos dados, que juntamente com estes processos metodológicos compreendidos das seções 3.1 e 3.10, compõem a metodologia aplicada para o cumprimento dos objetivos propostos neste estudo.

Berry (2004, p. 55) ainda denota que “Mineração de Dados não é um processo linear”.

A Figura 06 apresenta as etapas metodológicas da Mineração de Dados.

FIGURA 06 – Processo não linear de Mineração de Dados



FONTE: Berry (2004, p. 55). Tradução feita pelo autor.

3.1 Identificação do problema de negócio

A primeira etapa a se executar, para transformar um problema comum em um problema de Mineração de Dados, é analisar qual técnica da Mineração de Dados a se utilizar entre as seis técnicas (classificação, estimativa, previsão, análise de afinidade, análise de agrupamentos ou sumarização). Por isso, deve-se verificar se o problema pode ser resolvido por uma técnica supervisionada ou por uma não supervisionada (Berry, 2004). O presente estudo enquadra-se na técnica de análise de afinidade.

Outro item de vital importância, para Berry (2004), no momento da decisão de que técnica de Mineração de Dados utilizar é ter presente para que os resultados serão utilizados.

3.2 Seleção de dados apropriados

Conforme Berry (2004), deve-se levar em conta alguns itens para a aplicação de alguma técnica de Mineração de Dados. Um exemplo é que novos dados podem ser inseridos na base de dados, porém não podem alterar os registros passados (históricos), obtendo-se assim resultados coerentes em cada vez que se aplicam estas técnicas.

Alguns problemas também apontados por Berry (2004) são que muitas organizações não possuem as bases de dados atualizadas ou possuem mais de uma base de dados, não tendo as informações centralizadas.

3.3 Conhecer os dados

Berry (2004, p. 67) destaca que a pessoa responsável por minerar os dados deve ter conhecimento da base de dados na qual está minerando. Isto tanto em nível da estrutura do banco de dados como em nível lógico do negócio em questão, para descarte de soluções geradas que não são boas.

3.4 Criação do modelo

Nesta etapa, cria-se o modelo da estrutura de dados que irá se utilizar para o processo como um todo. Neste momento analisa-se a necessidade de criação de algum(ns) atributo(s) específico(s), que poderia ser resultado da combinação de dados da base de dados, por exemplo. Este modelo será utilizado em todas as etapas da mineração de dados (Berry, 2004).

Como exemplo, se a base de dados possui informações dos clientes, cada dado do cliente pode ser utilizado no modelo, conforme o objetivo que se necessita, tanto para constar na entrada de dados como para constar nos resultados gerados.

Se desejar informar a “idade média dos clientes”, por exemplo, tem-se que é necessário como entrada do sistema o dado referente a “data de nascimento” ou qualquer outro dado que possa mensurar o tempo de vida da pessoa, para no final do processo, gerar o resultado “idade média dos clientes”, por exemplo.

3.5 Resolver problemas dos dados

Berry (2004) salienta que todos os dados têm problemas. Por esta razão, tem-se que reparar os problemas dos dados antes de aplicar-se qualquer técnica de Mineração de Dados, para evitar problemas futuros nos resultados.

Berry (2004, p. 72) cita: “Todos os dados são sujos. Todos dados têm problemas”.

Outro grande cuidado que se devem ter ao analisar os dados, a fim de evitar conclusões erradas após a Mineração de Dados, segundo Berry (2004), é que o mesmo dado, ao ser coletado de fontes diferentes, pode estar diferente.

3.6 Transformação dos dados para construção de informações para mineração dos dados

Nesta etapa do processo, conforme Berry (2004), é feita a análise dos atributos que serão utilizados para o estudo. Se os atributos já estão corretos na base de dados, não é necessário nenhum trabalho adicional, senão é necessária uma transformação destes atributos. Como atributos possíveis, cita-se como exemplo “sexo”, “cor do cabelo”, “cor dos olhos”, “índice de obesidade”, etc. Neste exemplo utilizado, imagina-se que não se tem o item “índice de obesidade” e este é um item que se quer como resultado, porém tem-se na base de dados a “altura” e o “peso” da pessoa. Assim, poder-se-ia criar antes da Mineração de Dados, um campo “índice de obesidade” que seria o resultado de cálculo entre outros campos já existentes na base de dados.

3.7 Construção do modelo

Berry (2004) explica que, nesta etapa, para cada técnica/ algoritmo específico, tem-se uma técnica determinada para a construção do modelo, visando à obtenção de um resultado satisfatório após a execução da Mineração de Dados.

3.8 Avaliação do modelo

Para Berry (2004), cada técnica possui uma avaliação específica de seu modelo. Porém, o que se deve avaliar para todos os modelos são quatro itens, em forma de questionamento:

- quão exato é o modelo?
- como o modelo descreve os dados observados?
- qual a confiabilidade nas previsões do modelo específico?
- quão compreensível é o modelo?

Ou seja, segundo BERRY (2004, p. 78), deve-se avaliar além dos quatro itens acima, itens específicos da técnica/ algoritmo escolhido.

3.9 Interpretando o modelo

Neste item, faz-se a conversão dos resultados da Mineração de Dados para resultados numéricos e descritivos, com o objetivo de tornar mais compreensíveis e legíveis os resultados. O processo, conforme Berry (2004), pode ser feito por intermédio de software ou pela pessoa que irá minerar os dados. Ele não é efetuado somente por software, devido à Mineração de Dados criar atributos novos durante o processo, dos quais não se tem uma descrição prévia do que são, necessitando de intervenção humana para facilitar o entendimento posterior.

3.10 Avaliação dos resultados

A avaliação dos resultados é feita no término do processo, quando a coerência ou não dos dados deve ser examinada com base na realidade, segundo Berry (2004).

3.11 Aplicação do algoritmo *a priori* da técnica de análise de afinidades da Mineração de Dados

Neste item analisam-se algumas características da técnica de “análise de afinidades” e apresenta-se o algoritmo *a priori* utilizado para a Mineração de Dados neste estudo.

3.11.1 Representação de dados para regras de associação

Segundo Larose (2005), existem dois tipos principais de representação dos dados: *utilizando dados transacionais* ou *dados tabulares*. No primeiro, utiliza-se a estrutura de dois campos para representar um registro, sendo um campo código e o outro conteúdo. A Figura 07 relata essa estrutura.

FIGURA 07 – Formato de dados transacionais

Código da Transação	Itens
1	Brócolis
1	Pimentão Verde
1	Milho
2	Aspargos
2	Pepino
2	Milho
3	Milho
3	Tomates
⋮	⋮
⋮	⋮

FONTE: Larose (2005, p.183). Tradução feita pelo autor.

Na representação de dados tabulares, os dados são representados com “0”, se não existe determinado dado no registro, e com “1”, se existe. Na Figura 08, pode ser observada esta relação.

FIGURA 08 – Representação tabular dos dados

Transação	Aspargos	Feijão	Brócolis	Milho	Pimentão Verde	Pepino
1	0	0	1	1	1	0
2	1	0	1	1	1	1
3	0	1	1	1	1	1
4	0	1	1	1	1	0
5	1	1	1	0	0	0
6	1	1	0	0	0	1
7	0	0	0	1	0	0
8	0	0	1	0	1	0
9	1	1	0	0	1	1
10	0	1	0	1	0	0
11	0	1	1	0	0	1
12	1	1	0	0	0	1
13	1	1	0	0	1	1
14	0	1	1	1	1	0

FONTE: Larose (2005, p. 183). Tradução feita pelo autor.

3.11.2 Características dos algoritmos de regras de afinidades

Um exemplo de como as técnicas de afinidade tratam as regras de associação pode ser entendido no seguinte exemplo:

Tem-se uma base de dados **D**, na qual há transações **T**, conforme representado na Figura 08. No caso, se um determinado item **X** ocorre na base de dados, então um item **Y** também ocorre com alguma relação a **X**. Aqui se observa uma relação normal.

O conceito de co-ocorrência parte do princípio de que, se tem a ocorrência de **X**, **Y** e **Z** em **D**, então **A** também ocorre, com alguma relação a **X**, **Y** e **Z**. Com isso, eliminam-se regras triviais, por exemplo, *se tem comida e bebida então comida*.

A representação de *se A, então B*, será representada por $A \Rightarrow B$.

Outro conceito importante para a técnica de análise de afinidades é o conceito de *suporte*, que representa a proporção de determinadas transações em uma base de dados **D**.

$$\text{suporte} = s = P(A \cap B) = \frac{\text{Número de transações entre A e B}}{\text{Total de número de transações}}$$

A *confiança* tem por objetivo medir a confiança que se pode ter no resultado atingido.

$$\text{confiança} = c = P(B | A) = \frac{P(A \cap B)}{P(A)} = \frac{\text{Núm. trans. contendo A e B}}{\text{Num. trans contendo A}}$$

Com isso, tem-se um suporte melhor na tomada de decisões e o analista preferirá as regras que possuem maior suporte e uma maior confiança.

No exemplo dirigido deste trabalho, quanto maior o suporte e a confiança melhor e mais concreto será o resultado. Por exemplo, para detectar-se a compra conjunta de dois produtos, tendo um suporte de 20% e uma confiança de 70% com relação a determinado produto, o analista pode tirar a conclusão de que, em 70% dos casos, de cada cinco pessoas, uma delas compra esses produtos juntos com relação a um determinado produto.

Por outro lado, no caso de uma análise sobre fraudes e terrorismo, conforme exemplo citado por Larose (2005), o objetivo seria diminuir ao máximo este suporte, mas mantendo uma confiança alta.

Outros dois conceitos utilizados nas técnicas de análise de afinidades são *itemset* e *k-itemset*. Por exemplo, o item “comer” é um *itemset*, o conjunto {comer, beber} é um *k-itemset*, neste caso *2-itemset*.

O *itemset* é o número de vezes que um determinado item ocorre em um conjunto de transações; já o *itemset freqüente* é o *itemset* que ocorre um número mínimo de vezes, com uma freqüência maior ou igual à freqüência mínima necessária informada pelo analista (representada pelo símbolo θ). Por exemplo, se o item comer ocorre 5 vezes e $\theta = 4$, o item “comer” é considerado freqüente.

A partir dessas definições, se tem os dois primeiros passos básicos para minerar dados utilizando técnicas de análise de afinidades:

1. encontrar todos os *itemsets* com freqüências maiores ou iguais a θ .
2. Gerar, posteriormente, as regras de associação satisfazendo o suporte e a confiança mínima.

3.11.3 Algoritmo *a priori*

Este algoritmo tem como principal característica o objetivo de priorizar a minimização do espaço de pesquisa de regras de associação na Mineração de Dados.

Abaixo consta a forma de trabalho do algoritmo *a priori*:

Se na base de dados existe um item **Z** que não é freqüente, ao adicionar-se um item **A**, ou qualquer outro item a **Z**, não se tornará o item **Z** mais freqüente. Analisando o exemplo ainda da junção de **Z** não freqüente com um elemento **A** freqüente, $Z \cap A$ não será freqüente.

Nos itens 3.11.3.1 e 3.11.3.2 analisa-se a forma de trabalho do algoritmo *a priori*.

3.11.3.1 Gerando os *itemsets* freqüentes

Para demonstrar o funcionamento da geração de todas as instruções de *itemsets* freqüentes, utiliza-se a Tabela 01 abaixo:

TABELA 01 – Lista de compras

Transação	Itens
1	Leite, ovos, fraldas, açúcar, café, manteiga
2	Farinha, café, leite
3	Açúcar, ovos, leite
4	Açúcar, café
5	Fraldas
6	Leite, ovos, manteiga
7	Ovos, café, leite, açúcar
8	Farinha, manteiga, ovos
9	Leite, café, açúcar, ovos, manteiga
10	Fraldas, cerveja, café

FONTE: Elaborada pelo autor

Posteriormente são gerados todos os *k-itemsets* necessários. Para a geração deste exemplo, são utilizados o suporte mínimo de 25% e a confiança de 75%.

A Tabela 02 exibe a lista de *1-itemsets*.

TABELA 02 - Lista de 1-itemsets

1-itemsets	Suporte
Leite	60%
Ovos	60%
Café	60%
Açúcar	50%
Fraldas	30%
Manteiga	40%
Farinha	20%
Cerveja	10%

FONTE: Elaborada pelo autor

Nota-se na Tabela 02 que os itens “Farinha” e “Cerveja”, não atingiram o suporte mínimo de 25% ou seja, serão desconsiderados nas análises de *k-itemsets* posteriores.

Na Tabela 03, segue a lista de *2-itemsets*.

TABELA 03 - Lista de 2-itemsets

2-itemsets	Suporte
Leite, ovos	50%
Leite, café	40%
Leite, açúcar	40%
Leite, fraldas	10%
Leite, manteiga	30%
Ovos, café	30%
Ovos, açúcar	40%
Ovos, fraldas	10%
Ovos, manteiga	40%
Café, açúcar	40%
Café, fraldas	20%

2-itemsets	Suporte
Café, manteiga	20%
Açúcar, fraldas	10%
Açúcar, manteiga	20%
Fraldas, manteiga	10%

FONTE: Elaborada pelo autor

Na Tabela 04 é apresentada a lista de 3-itemsets, eliminando-se os itens com suporte inferior a 25%, constantes na Tabela 03.

TABELA 04 - Lista de 3-itemsets

3-itemsets	Suporte
Leite, ovos, café	30%
Leites, ovos, açúcar	40%
Leite, ovos, manteiga	30%
Leite, café, açúcar	30%
Leite, café, manteiga	20%
Leite, açúcar, manteiga	20%
Ovos, café, açúcar	30%
Ovos, manteiga, café	20%
Ovos, açúcar, manteiga	20%
Café, açúcar, manteiga	20%

FONTE: Elaborada pelo autor

Na Tabela 05, mostra-se a lista de 4-itemsets, tendo sido desconsiderados os itens com suporte inferior a 25% da Tabela 4.

TABELA 05 - Lista de 4-itemsets

4-itemsets	Suporte
Leite, ovos, café, açúcar	30%
Leite, ovos, manteiga, café	20%
Leite, café, açúcar, manteiga	20%
Leite, açúcar, manteiga, ovos	20%
Ovos, café, açúcar, manteiga	20%

FONTE: Elaborada pelo autor

Neste processo não é possível gerar uma lista com 5-itemsets, pois não existem mais ocorrências na base com estas características.

O item 3.11.3.2 aborda as regras de associação criadas, referenciando a confiança dos itens.

3.11.3.2 Geração de regras de associação

Neste item são geradas as regras de associação para análise da confiança dos resultados, visando a elaboração do resultado final da Mineração de Dados. Esta etapa do algoritmo *a priori* é dividida em duas sub-etapas:

- geração de todos os relacionamentos, conforme o item 3.11.3.1;
- cálculo da confiança destes itens focando o cumprimento da confiança mínima necessária.

Como exemplo, pode-se citar o item abaixo, junto com os cálculos:

“Se compra leite e ovos, então compra açúcar”.

$$\text{suporte} = \frac{4}{10} = 40\%$$

$$\text{confiança} = \frac{4}{5} = 80\%$$

Com isso, tem-se o suporte e a confiança do(s) item(s) referenciado(s).

Alguns analistas utilizam para a análise a multiplicação do suporte pela confiança para cada regra de associação.

4 CARACTERIZAÇÃO DA EMPRESA

Nesta seção é feita breve caracterização da empresa na qual a ferramenta desenvolvida foi aplicada para fins deste estudo.

O estabelecimento estudado foi inaugurado no ano de 1980, no Rio Grande do Sul, em município no qual está funcionando até a data atual.

Quando a empresa surgiu, a idéia era trabalhar apenas no segmento de padaria, porém, com o passar do tempo, viu-se a necessidade de implantar algum diferencial diante de seus concorrentes. Com esse objetivo, o proprietário implantou a venda de produtos alimentícios.

A empresa conta com sete funcionários e também, com dois sócios que atuam na mesma. Comercializa mais de 3.000 itens distribuídos em oito linhas de produtos (calçados, bebidas, padaria, fruteira, limpeza e higiene, produtos alimentícios, material escolar e medicamentos).

5 DESENVOLVIMENTO

Nesta etapa são descritos os passos que foram necessários para o desenvolvimento da ferramenta de Mineração de Dados utilizando a técnica de análise de afinidades com o algoritmo *a priori*.

5.1 Análise dos dados da empresa

Como descrito na seção 3.2 do capítulo 3, uma das primeiras etapas para o desenvolvimento de um programa/ferramenta de Mineração de Dados é selecionar os dados, ou a base de dados na qual se objetiva minerar os dados na busca de algum padrão não conhecido.

Analisando a estrutura da base de dados utilizada pela empresa, constata-se que o uso de apenas duas tabelas de sua base é suficiente para a obtenção dos resultados visados, utilizando a técnica de Mineração de Dados e algoritmo referidos.

Para o correto funcionamento da ferramenta desenvolvida, é preciso que os dados aos quais se deseja aplicar o algoritmo estejam em um arquivo, tabela ou qualquer outra estrutura de dados que possibilite a exportação para um arquivo no formato CSV¹.

¹CSV: Sigla para *comma-separated values*, que é um formato de arquivo que utiliza vírgula para separar valores em uma linha, quebra de linha para separar linhas e ainda aspas duplas em campos textuais que possuem vírgula e quebra de linha.

São necessários também dois arquivos CSV para o correto funcionamento da ferramenta desenvolvida: um arquivo contendo os registros das compras efetuadas no estabelecimento e o outro com o cadastro dos produtos disponíveis da compra. A estrutura desses arquivos é descrita nas seções 5.1.1 e 5.1.2.

5.1.1 Registro de compras

Nesta seção descreve-se a estrutura do arquivo utilizado para registrar as compras efetuadas no estabelecimento para análise com a ferramenta desenvolvida. Este arquivo deve obedecer às seguintes exigências:

- cada linha do arquivo deve conter um item adquirido do estabelecimento por um cliente;
- possuir uma coluna que corresponde ao código do produto/item adquirido;
- ter uma coluna que referencie a compra em si, e não o produto. Por exemplo, o número da nota fiscal. Este dado é muito importante, pois se uma pessoa efetua a compra de vários produtos em uma única vez, ela terá uma linha registrada neste arquivo para cada produto adquirido, independente da quantidade, e precisará ter um dado que será o mesmo para todas as linhas dos produtos adquiridos, que representará a compra em si. Um detalhe importante sobre esta referência é que ela deve ser única para cada compra, ou seja, se determinada pessoa efetua uma compra em um dia, o código de sua compra será um para os “n” produtos adquiridos. Se a mesma pessoa fizer compras também no dia seguinte no mesmo estabelecimento, o código de referência tem que ser diferente do da compra do dia anterior.

Abaixo segue exemplo de estrutura de um arquivo CSV (Tabela 06) que cumpre corretamente com as exigências da ferramenta desenvolvida:

TABELA 06 - Exemplo de estrutura de arquivo CSV para registro de compras

Coluna 1	Coluna 2	Coluna 3	Coluna 4	Coluna N
1234		66678		
1234		99900		
1234		9807		
3456		9807		

Coluna 1	Coluna 2	Coluna 3	Coluna 4	Coluna N
8888		1234		
8888		8798		

Fonte: Elaborada pelo autor.

No exemplo proposto na Tabela 06, foram efetuadas três compras, representadas pelos códigos de compra 1234, 3456 e 8888. Os produtos comprados na compra de número 1234 foram os de código: 66678, 99900 e 9807; o produto adquirido pela compra 3456 foi o de código 9807 e os obtidos na compra de código 8888 foram 1234 e 8798.

Destaca-se ainda, que as colunas 2, 4 e 5, não possuem nenhum dado. Esta estrutura é apresentada propositalmente desta forma, porque o arquivo pode ter “n” colunas, com qualquer valor, sendo necessário apenas que entre estas estejam presentes duas colunas, uma representando um identificador único de compra e a outra o produto/item adquirido. O número da coluna onde está presente cada item poderá ser informado pelo usuário.

No exemplo demonstrado na Tabela 06, a coluna 1 seria informada na ferramenta como sendo a que contém a informação referente ao código de compra; já a coluna 3 seria identificada como a que dispõe da informação relativa ao produto adquirido. O mesmo exemplo poderia ter a data de compra, quantidade comprada, valor do item etc. dispostos nas colunas que estão com os dados em branco.

5.1.2 Registro de produtos

Neste item é apresentada a estrutura necessária do arquivo a ser utilizado na importação das informações referentes aos produtos cadastrados que se encontram disponíveis no estabelecimento.

O arquivo que contém o cadastro de produtos deve obedecer às seguintes exigências:

- cada linha do arquivo contém um produto cadastrado e disponível no estabelecimento;

- a linha tem que possuir uma coluna que referencie o código do produto, que deve ser unívoco, não podendo existir mais de um produto diferente com o mesmo código. Este código é o mesmo utilizado no arquivo de registro das compras efetuadas e registradas no arquivo de registro de compras (descrito na seção 5.1.1).

- coluna que informe a descrição e/ou qualquer outro item que desejar informar no momento de apresentação dos resultados. Este item é de fundamental importância, pois será o mostrado na totalização e análise dos resultados propostos para o analista.

Abaixo segue um exemplo de arquivo CSV (Tabela 07) que cumpre corretamente com as exigências da ferramenta desenvolvida:

TABELA 07 – Exemplo de estrutura de arquivo CSV para registro de produtos

Coluna 1	Coluna 2	Coluna 3	Coluna 4	Coluna N
	66678		Abacaxi	
	99900		Chocolate	
	9807		Alface	
	1234		Pão	
	8798		Fermento	

Fonte: Elaborada pelo autor.

No exemplo proposto na Tabela 07 são efetuados cinco cadastros de produtos representados pelos códigos de produto 66678, 99900, 9807, 1234 e 8798. Para cada produto também foi efetuado um cadastro de descrição do produto conforme coluna 4, ou seja, cadastro de informação que é apresentada na análise dos resultados. Cada descrição informada na coluna 4 corresponde a um código de produto informado na coluna 2.

Assim como no registro de compras citado na seção 5.1.1, no registro de produtos também são mostradas colunas em branco, ou seja, colunas que poderiam possuir qualquer dado, sendo necessário apenas uma coluna que identifique o código do produto e uma que identifique a descrição a ser mostrada quando este código de produto for referenciado na análise dos resultados gerados pela ferramenta.

Posteriormente a esta etapa de seleção dos dados adequados para uma mineração de dados, o ideal é efetuar-se pré-processamento dos dados.

5.2 Pré-processamento dos dados

Esta etapa é indicada, como mencionado na seção 2.1, para efetuar a limpeza dos dados antes do processamento que utiliza técnicas específicas aplicadas para a descoberta de conhecimento em base de dados, como a Mineração de Dados. Neste sentido, esta técnica envolve maior análise, cadastros corretos dos dados, registro correto de compras, etc., do que processamento via software para esta limpeza.

No contingente de dados analisados e processados neste trabalho, estão mais de 5.000 compras e um total de mais de 9.000 produtos existentes na base de dados de estudo. Em uma análise preliminar da base de dados da empresa, verificou-se que existem algumas inconformidades nos dados, como: mesmo produto cadastrado com códigos diferentes; produtos cadastrados mais de uma vez, com pequenas alterações em sua descrição, como utilização ou não de acentuação gráfica; espaçamento entre nomes etc. Esses problemas não puderam ser corrigidos por demandarem muito trabalho de análise e ajustes e não haver tempo hábil de aplicação. Porém, é observado no capítulo 7, referente à conclusão do trabalho, como sendo uma sugestão para a empresa analisada o ajuste dessas inconformidades, visando melhor resultado do sistema.

Outro tipo de problema encontrado neste pré-processamento dos dados foi o cadastro de vários produtos com um código “curinga”. Isso ocorria porque por algum motivo o atendente ao registrar a compra não registrava o código do produto, e sim, este código “curinga”, que lhe abria a opção de registrar o preço do produto.

Para a minimização dos problemas, na ferramenta proposta, utilizou-se o registro de produtos nas tabelas auxiliares utilizadas pela ferramenta, com o registro destes códigos como sendo campos “textuais” ao invés de campos “numéricos”, solução esta que se enquadra no pré-processamento de dados. Como exemplo de um problema encontrado tem-se a existência de produtos cadastrados como código “000007” e produtos cadastrados como código “7”, os quais se equivalem mas referem-se a produtos diferentes. Outro problema que ocorreria, se não fosse utilizada a solução apontada, é na existência de produtos com código “0” que não seriam considerados.

5.3 Estrutura de desenvolvimento

Nesta seção é abordada a estrutura de desenvolvimento utilizada, bem como algumas informações técnicas sobre o desenvolvimento.

Um fator importantíssimo na estrutura do desenvolvimento é a decisão de armazenar e utilizar banco de dados, tanto para registro das compras como para registro dos resultados. Esta decisão tem grande relevância, pois torna a ferramenta mais eficiente nas análises posteriores à geração das informações. Isto, porque os processos que demandam mais tempo na ferramenta são os de registro de compras, registro de produtos e em especial a Mineração de Dados em si. Esses processos não são feitos diariamente, somente são executados quando o analista necessitar gerar resultados com dados que não estavam presentes na análise anterior. Porém, a “Visualização dos Resultados” (que é uma opção da ferramenta desenvolvida) poder ser feita quantas vezes se considerar adequado, demandando este processo tempo irrisório na sua execução em comparação com os anteriores.

O desenvolvimento desta ferramenta é proposto em três etapas, como se demonstra na Figura 10.

FIGURA 09 - Estrutura e desenvolvimento da ferramenta



Fonte: Elaborada pelo autor.

Um fator muito importante encontra-se já na primeira etapa desta estrutura de desenvolvimento proposta. Devido à existência de muitos bancos de dados diferentes, tendo cada um suas particularidades, funcionalidades e muitos outros fatores, optou-se pela efetuação da entrada dos dados via arquivo CSV. Com isso, tem-se grande independência do banco de dados, os únicos gerenciadores de banco de dados que não se enquadrariam, seriam os que não exportam diretamente para formato CSV.

Essa escolha deu-se principalmente pelo CSV ser um formato de texto “limpo” que possui poucas codificações se comparado com outros formatos. Outro fator

importante foi o de que muitos sistemas estão desenvolvidos utilizando sistemas gerenciadores de banco de dados que não suportam a exportação de outros formatos que se apresentam como tendência nos dias de hoje.

Analisando a primeira etapa demonstrada na Figura 10, a ferramenta tem que ser alimentada com um arquivo CSV que contenha o registro das compras efetuadas no estabelecimento estudado. Também nesta etapa deve-se realizar o cadastro de produtos, que é feito com a importação dos dados de um arquivo em formato CSV.

Posteriormente, no mesmo processo, chamado na ferramenta de “Leitura de Compras” e “Leitura de Produtos”, estes dados dos arquivos CSV são armazenados em tabelas auxiliares em um gerenciador de banco de dados Firebird. Essa escolha decorre do fato de o autor já possuir familiaridade com o banco de dados, e o mesmo se encontrar como “Software Livre”, o que permite utilizá-lo sem restrições legais.

Na segunda etapa, após os dados terem sido armazenados, pode-se efetuar o processo de Mineração de Dados utilizando a técnica de análise de afinidades com o algoritmo *a priori*.

Para esta segunda etapa, o analista ou a pessoa que estiver executando o software tem que informar primeiramente o item freqüente que deseja ou o suporte mínimo ao qual o algoritmo tem que levar em conta para o seu processamento. Após receber estas informações, o algoritmo inicia a execução efetuando as análises necessárias e gerando os resultados em tabelas do gerenciador de banco de dados Firebird.

A etapa de visualização de resultados (terceira) é a que, em uma primeira execução, deve acontecer em momento posterior à primeira e segunda etapas, porém, após uma vez executado o processo inicial, ela pode acontecer independente das duas anteriores. Nesta etapa, os resultados podem ser visualizados pelo analista ou por qualquer outra pessoa que possuir acesso ao software.

5.4 Ambiente de desenvolvimento

Para a escolha do ambiente ao qual seria desenvolvida esta ferramenta, levou-se em conta alguns fatores primordiais: possibilidade de utilização com maior independência possível da base de dados utilizada pelo estabelecimento; linguagem que possibilita a integração com algum banco de dados robusto, que suporte grande quantidade de dados, e, o mais importante, familiaridade e condições de o autor efetuar o desenvolvimento na plataforma escolhida.

Devido aos dois primeiros itens serem muito subjetivos (independência de base de dados e robustez de banco de dados), optou-se por maior ênfase no terceiro item (familiaridade do autor). Então se optou pela linguagem Pascal com ambiente de desenvolvimento Delphi (versão 6.163) e sistema gerenciador de banco de dados Firebird 2.0.

6 ANÁLISE DOS RESULTADOS

Neste capítulo é abordada a análise dos resultados gerados pela ferramenta quando aplicada à base de dados do estabelecimento em estudo.

6.1 Base de dados

Atualmente a empresa possui 9.994 produtos diferentes cadastrados no sistema. A base de dados utilizada possui o registro de todas as compras feitas no estabelecimento entre o dia 02 de fevereiro de 2007 e 19/08/2007. No período, foi efetuada a venda de 71.553 produtos, independente de sua quantidade. Estes produtos foram vendidos em 5.165 cupons de vendas (compras). Com isso, dividindo o número de produtos vendidos pela quantidade de compras efetuadas, tem-se que a média de produtos adquiridos por compra é de catorze produtos (a média exata seria de 13,85, porém, como não existe a venda de produtos “quebrados”, como 0,8 ou 0,5 produtos, arredondou-se o número para catorze produtos em média).

6.2 Mineração de Dados no estabelecimento

Para a mineração de dados e geração dos resultados para análise, utilizou-se o suporte mínimo de 5%. É possível utilizar um percentual maior ou menor, ficando a cargo do analista escolher o que deseja analisar. Salienta-se, porém, que quanto menor for o percentual maior será o tempo de processamento dos resultados. Por exemplo, se o analista optar pelo percentual de 0%, analisará os dados com todos os itens que possuem 0% de vendas ou mais, ou seja, utilizaria todos os itens, obtendo o mesmo resultado de uma análise de afinidades simples, sem aplicação do algoritmo *a priori*, ou seja, levaria tempo excessivo para gerar o resultado. Esse ponto já foi salientado na seção 2.5.7.

6.3 Resultados no estabelecimento

Após os dados serem processados, é possível gerar os resultados com o auxílio da ferramenta desenvolvida, clicando no botão “Visualização dos Resultados” (anexo E). Abre uma tela na qual é informada a confiança mínima que é desejada nos resultados. Para analisar os resultados neste trabalho, foi atribuída confiança mínima de 25%. Outro item que se destaca é que nesta lista só aparecem os dados em que a confiança mínima é atingida.

Após a obtenção dos resultados com esta confiança mínima (os resultados na sua totalidade estão no Anexo F), fazem-se algumas análises de compras para demonstrar como os dados devem ser analisados. Destaca-se que o algoritmo analisa compras em conjunto, ou seja, dois produtos vendidos juntamente, três produtos vendidos juntamente e assim por diante, desde que seja satisfeito o suporte mínimo para a geração e a confiança mínima no resultado. Abaixo seguem algumas análises:

- “Brilhante em Pó em 1 Kg” e “Produtos Diversos Balança”: possui suporte de 3,12%, ou seja, de cada cem compras efetuadas no estabelecimento, três compras possuem os produtos “Brilhante em Pó em 1 Kg” junto com “Produtos Diversos Balança”. Outra análise possível de fazer nesta mesma compra, é a de confiança que, para este resultado, analisando pelo produto “Brilhante em Pó em 1

Kg” é de 25,55%, o que significa que uma em cada quatro pessoas que adquirem “Brilhante em Pó em 1 Kg” também compra “Produtos Diversos Balança”.

- “Café Fritz Frida 200g” e “Óleo de Soja Primor”: tem suporte de 2,34%, o que representa que de cada cem pessoas que fazem compras no estabelecimento, duas adquirem esses produtos juntamente. Ainda destaca-se 42,45% de confiança com relação ao “Café Fritz Frida 200g”, de cada cem pessoas que compram o “Café Fritz Frida 200g” 42 também adquirem o “Óleo de Soja Primor”.

- “Brilhante em Pó em 1 Kg” e “Óleo Soja Primor”: possui suporte de 4,84%, o que aponta que de cada cem compras no estabelecimento, quatro incluem o produto “Brilhante em Pó em 1 Kg” em conjunto com o “Óleo Soja Primor”. Assim, observa-se a confiança mínima de 39,68% com relação ao “Brilhante em Pó em 1 Kg”. Sendo que das pessoas que compram “Brilhante em Pó em 1 Kg”, 39,68% também adquirem o “Óleo Soja Primor”. Agora, analisando a confiança com relação ao “Óleo Soja Primor”, tem-se o índice de 27,59%, que refere que, das pessoas que compram “Óleo Soja Primor” 27,59% também adquirem o “Brilhante em Pó em 1 Kg”.

- “Fermento Saft-Instant” e “Óleo Soja Primor”: suporte de 2,94%, ou seja, de cada cem pessoas que compram no estabelecimento, duas adquirem “Fermento Saft-Instant” juntamente com “Óleo Soja Primor”. Saliencia-se ainda a confiança de 45,10% com relação ao produto “Fermento Saft-Instant”. Portanto, de cada cem pessoas que adquirem o produto “Fermento Saft-Instant” 45 adquirem o produto “Óleo Soja Primor” juntamente.

- “Vinagre Branco Prinz” e “Óleo Soja Primor”: suporte de 3,00%. O dado revela que de cada cem compras no estabelecimento, três compras incluem os produtos “Vinagre Branco Prinz” e “Óleo Soja Primor”. Constata-se ainda a confiança de 46,26% em relação ao produto “Vinagre Branco Prinz”, de cada cem pessoas que adquirem o produto “Vinagre Branco Prinz” 46 também adquirem o produto “Óleo Soja Primor”.

- “Feijão Preto Fritz e Frida” e “Óleo de Soja Primor”: suporte de 2,94%, o que representa que duas em cada 100 compras são de “Feijão Preto Fritz e Frida” e “Óleo de Soja Primor” juntamente. Com relação à confiança, esta é de 48,87% com relação ao produto “Feijão Preto Fritz e Frida”. Em outras palavras, de cada cem

peças que compram “Feijão Preto Fritz e Frida”, 48 adquirem também “Óleo de Soja Primor”. Salienta-se ainda, que se o índice fosse de 50% (de cada cem compras cinquenta eram de determinado produto), pode-se analisar que de cada duas compras uma é do produto analisado.

- “Fermento Saft-Instant” e “Erva Amável”: suporte de 2,71%. O índice representa em cada cinquenta compras, uma é destes produtos juntos. Aqui se observa confiança de 41,54% para o produto “Fermento Saft-Instant” de cada cem compras do produto “Fermento Saft-Instant” 41 também adquirem o produto “Erva Amável”.

- “Cebola” e “Batata”: suporte de 5,32%, o que significa que em cinco de cada cem compras no estabelecimento analisado inclui esses produtos em conjunto. Este dado também pode ser analisado como de cada vinte compras efetuadas, uma compra contém os produtos “Cebola” e “Batata” juntamente. Com relação à confiança, o produto “Cebola” apresenta confiança de 50,73%, ou seja, de cada duas pessoas que compram “Cebola”, uma também compra “Batata”. Já a análise com relação ao produto “Batata”, apresenta índice de confiança de 38,84%, que representa que de cada cem pessoas que compram “Batata”, 38 também adquirem “Cebola”.

- “Batata”, “Erva Amável” e “Óleo de soja Primor”: suporte de 1,99%, isto é de cada cem pessoas duas adquirem os três produtos juntamente. Destaca-se o índice de confiança de 33,88% para “Erva Amável” e “Óleo de Soja Primor” de cada cem pessoas que adquirem “Erva Amável” e “Óleo de Soja Primor” 33 pessoas também compram “Batata”.

- “Óleo de Soja Primor”, “Cebola” e “Batata”: suporte de 2,57%, em cada cem pessoas duas contém produtos “Óleo Soja Primor”, “Cebola” e “Batata” juntamente. Neste item tem-se confiança de 48,36% para os produtos “Cebola” e “Batata”, pois de cada cem pessoas que adquirem estes produtos, 48 também compram “Óleo de Soja Primor”.

- “Tomate”, “Batata” e “Óleo de Soja Primor”: suporte de 1,80%, portanto, de cada cem pessoas que compram no estabelecimento, uma adquire os produtos “Tomate”, “Batata” e “Óleo de Soja Primor” juntamente. Analisando a confiança,

destaca-se o índice de 35,61% com relação aos produtos “Batata” e “Óleo de Soja Primor”, o que indica que de cada cem pessoas que adquirem “Batata” e “Óleo de Soja Primor”, 35,61% também compram “Tomate”.

De posse dos dados gerados pela ferramenta desenvolvida, o analista pode examiná-los e traçar estratégias de disposição de produtos no supermercado, organização de “kits” com desconto de produtos e outras possibilidades de negócio que considerar ser ideal.



7 CONSIDERAÇÕES FINAIS

A análise de dados é cada vez mais importante para as empresas nos dias de hoje, tornando-se um diferencial nos negócios as organizações que conseguem transformar dados em informações úteis e estratégicas.

Ao mesmo tempo, é necessária uma mão de obra qualificada e ferramentas que propiciem trabalho automatizado sobre este contingente de dados para transformá-los em informações úteis para a tomada de decisões pelos analistas.

Com o foco na transformação de dados existentes em empresas, este trabalho estudou a área de Mineração de Dados e a aplicou por meio da implementação de uma ferramenta para estabelecimentos comerciais por meio da técnica de análise de afinidades.

A ferramenta propiciou a empresa na qual foi aplicada disponibilidade de informações sobre os dados que já eram registrados normalmente que antes não eram percebidas. Essas informações poderão ser utilizadas visando ao aumento de vendas, melhor disposição de produtos no estabelecimento, formação de kits de vendas com desconto aos consumidores e outras aplicações criadas pela gerência do estabelecimento.

Um fator também destacado e sugestionado ao estabelecimento é o de realizar conferência e reorganização de cadastros de produtos, pois alguns se

encontram com código duplicados e há produtos idênticos com mais de um código cadastrado. Também neste sentido, outra sugestão feita a empresa é de preparar melhor seus atendentes e até rever o sistema de codificação de produtos, visando à utilização cada vez menor de códigos “curingas”, situação em que o atendente informa este código e posteriormente informa o valor da compra, não efetuando o processo correto que seria o registro específico do produto, através de seu código.

Considera-se, desta forma, que o trabalho atingiu o seu objetivo principal e os objetivos específicos definidos. Respondeu, portanto, também ao problema proposto. Destacou-se a área de descoberta de conhecimento em base de dados, focando na etapa de Mineração de Dados a qual foi foco principal deste trabalho. Nesta parte de Mineração de Dados, com base bibliográfica, estabeleceu-se que a área de aplicação da ferramenta a ser desenvolvida seria a análise de afinidades com o algoritmo *a priori*, o qual atingiria os objetivos deste trabalho.

Beneficia-se o pesquisador, que adquiriu conhecimento mais profundo em uma área essencial, em seu ponto de vista, ao futuro das organizações; o estabelecimento comercial, que dispõe de informações que podem lhe gerar diferencial competitivo tanto em aumento de vendas brutas como em maior lucratividade em sua área de atuação; o Centro Universitário UNIVATES, que, sendo uma instituição de ensino, disponibilizará um trabalho aplicado na área de mineração de dados para futuras consultas de seus alunos.

Para possíveis pesquisas e/ou trabalhos futuros nesta área, sugere-se a aplicação de outras técnicas de Mineração de Dados, para problemas semelhantes, ou até para problemas diferentes, visando a agregar maior número de técnicas dessa área aplicada. Também se sugere a elaboração de trabalhos futuros em técnicas de descoberta de conhecimento em base de dados, como exemplo, trabalhos que implementem técnicas de pré-processamento inteligente de dados, não sendo necessário eliminar duplicidades e dados afins manualmente.

8 REFERÊNCIAS BIBLIOGRÁFICAS

BERRY, Michael J. A.. **Data mining techniques: for marketing, sales and customer relationship management**. Indianapolis : Wiley Publishing, Inc, 2004.

CARVALHO, Juliano V., SAMPAIO, Marcus C., MONGIOVI, Giuseppe. **Utilização de Técnicas de “Data Mining” para o Reconhecimento de Caracteres Manuscritos**. Paraíba.

CARVALHO, Luís A. V. de. **Datamining: a Mineração de Dados no marketing, medicina, economia, engenharia e administração**. São Paulo : Érica, 2001.

CUNICO, Luiz H. B.. **Técnicas em Data Mining Aplicadas na Predição de Satisfação de Funcionários de uma Rede de Lojas do Comércio Varejista**. Curitiba, 2005. Tese de Mestrado, Setores de Tecnologia e de Ciências Exatas, Universidade Federal do Paraná.

FAYYAD, Usama M., PIATETSKY-SHAPIO, G., SMYTH, P., UTHURUSAMY, R.. **Advances in knowledge Discovery & Data Mining**. AAAI/MIT, 1996.

HAND, David, MANNILA, Heikki, SMYTH, Padhraic,. **Principles of Data Mining**. London, England : Bradfort Book, 2001.

LAROSE, Daniel T.. **Discovering knowledge in data: an introduction to data mining**. New Jersey : John Wiley & Sons, Inc. United States, 2005.

LEMOES, Eliane P.. **Análise de Crédito Bancário com o uso de Data Mining: Redes Neurais e Árvores de Decisão**. Curitiba, 2003. Tese de Mestrado, Setores de Tecnologia, Departamento de Construção Civil e de Ciências Exatas, Universidade Federal do Paraná.

LIMA, Paulo R. L.. **Um Sistema utilizando Data Mining para o Zoneamento e Monitoramento Agrícola no Brasil: Um caminho para a qualidade de software**. Florianópolis, 2001. Tese de Mestrado, Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina.

LOSHIN, David. **Business Intelligence The Savvy Manager's Guide**. San Francisco : Morgan Kaufmann Publishers, 2003.

REZENDE, Solange O. **Sistemas inteligentes : fundamentos e aplicações**. Barueri, SP : Manole, 2005.

ROCHA, Armando, MARTINS, D. B. M., COUTO, Daniel C., ROCHA, Rogério. **Conceitos Básicos sobre Data Mining**. Salvador, 2003.

RUD, Olívia Parr. **Data Mining Cookbook Modeling Data for Marketing, Risk, and Customer Relationship Management**. New York : John Wiley & Sons, Inc. United States, 2001.

SERRA, Laércio. **A essência do Business Intelligence**. São Paulo : Berkeley Brasil, 2002.

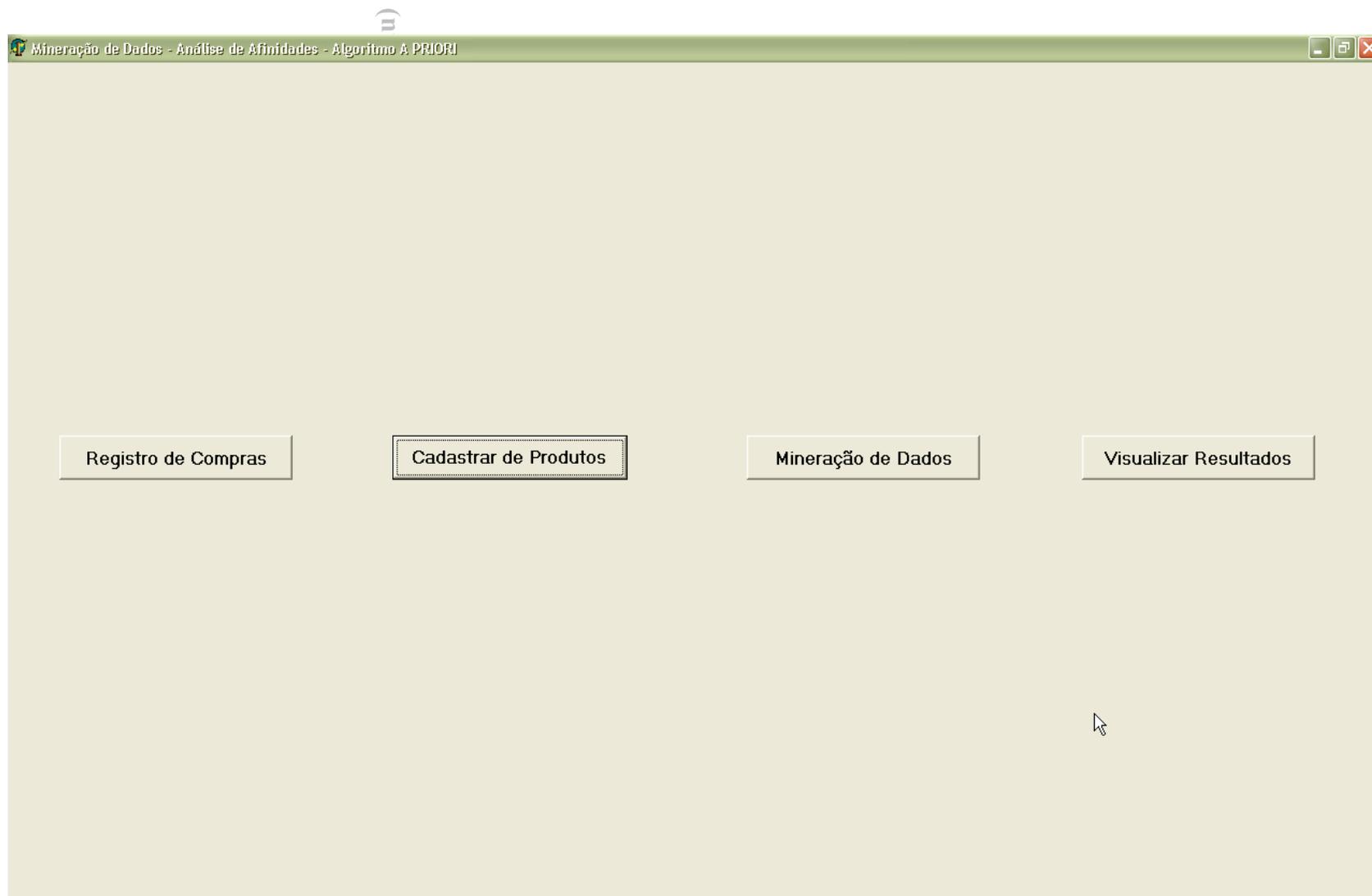
WITTEN, I. H.. **Data mining: practical machine learning tools and techniques**. San Francisco : Elsevier, 2005.



LISTA DE ANEXOS

ANEXO A – Tela inicial da ferramenta desenvolvida	65
ANEXO B – Tela para registro de compras na ferramenta desenvolvida.....	66
ANEXO C – Tela para cadastrar produtos na ferramenta desenvolvida	67
ANEXO D – Tela para executar processo de mineração de dados na ferramenta desenvolvida	68
ANEXO E – Tela para “Visualizar Resultados” na ferramenta desenvolvida	69
ANEXO F – Resultados da Mineração de Dados com confiança mínima de 25%	70
ANEXO G – Código de Implementação do formulário principal	87
ANEXO H – Código de Implementação do formulário de “Registro de Compras”.....	88
ANEXO I – Código de Implementação do formulário de “Cadastrar Produtos”	94
ANEXO J – Código de Implementação do formulário de “Mineração de Dados”	98
ANEXO K – Código de Implementação do formulário de “Visualizar Resultados” ..	114

ANEXO A – Tela inicial da ferramenta desenvolvida



ANEXO B – Tela para registro de compras na ferramenta desenvolvida

Registro de Compras

Neste formulário é feita a leitura de um arquivo em formato CSV para registro das compras do estabelecimento.

Número da Coluna que Corresponde ao **Identificador da Compra**: **Informe apenas números inteiros neste campo.**

Número da Coluna que Corresponde ao **Código do Produto Adquirido**: **Informe apenas números inteiros neste campo.**

Logs

- NÃO mostrar Log de Registro (Processo mais RÁPIDO, sem um acompanhamento do processo)
- Mostrar Log de Registro (Processo mais LENTO mas gera um acompanhamento do processo)

*** Registrar Compras ***

✓ Salvar Log

Fechar

ANEXO C – Tela para cadastrar produtos na ferramenta desenvolvida

Cadastro de Produtos

Neste formulário é feita a leitura de um arquivo em formato CSV para registro dos produtos do estabelecimento.

Número da Coluna que Corresponde ao **Código do Produto**: **Informe apenas números inteiros neste campo.**

Número da Coluna que Corresponde a **Descrição do Produto**: **Informe apenas números inteiros neste campo.**

Logs

NÃO mostrar Log de Registro (Processo mais RÁPIDO, sem um acompanhamento do processo)

Mostrar Log de Registro (Processo mais LENTO mas gera um acompanhamento do processo)

***** Registrar Produtos *****

ANEXO D – Tela para executar processo de mineração de dados na ferramenta desenvolvida

Mineração de Dados

Neste formulário é feita a Aplicação da Técnica de Análise de Afinidades com o algoritmo A PRIORI.
Informe um Item Frequente ou um Suporte Mínimo a ser utilizado pelo algoritmo. Se informar ambos, o algoritmo utilizará somente o SUPORTE MÍNIMO.

Item Frequente: **Informe neste campo a quantidade mínima de ocorrência compras registrada para um produto ou para um conjunto de produtos**

Suporte Mínimo: % **Informe neste campo o suporte mínimo desejado. (Utilize os números de 0 a 100, representando percentual de suporte mínimo)**

Logs

NÃO mostrar Log de Registro (Processo mais RÁPIDO, sem um acompanhamento do processo)

Mostrar Log de Registro (Processo mais LENTO mas gera um acompanhamento do processo)

***** Minerar Dados *****

Salvar Log

Fechar

ANEXO E – Tela para “Visualizar Resultados” na ferramenta desenvolvida

The screenshot shows a window titled "Resultados" with a standard Windows-style title bar. The main content area contains the following elements:

- A header text: "Neste formulário apresenta-se os resultados da Mineração de Dados. Os resultados são apresentados na planilha abaixo:"
- A label "Confiança Mínima" followed by an empty text input field.
- A yellow-highlighted instruction: "Informe apenas números inteiros neste campo. Aqui, é informada a confiança mínima para apresentação dos resultados."
- A button labeled "*** Gerar Resultado ***" with a mouse cursor hovering over it.
- A large empty table area with a header row and a highlighted green row.
- A footer logo for "Eechar" at the bottom center.

ANEXO F – Resultados da Mineração de Dados com confiança mínima de 25%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
FERMENTO SAFT-INSTANT , PRODUTOS DIVERSOS BALANCA	91	1,76185870170593%		
			FERMENTO SAFT-INSTANT	27,0029678344727%
CAFE FRITZ FRIDA 200G , PRODUTOS DIVERSOS BALANCA	73	1,41335916519165%		
			CAFE FRITZ FRIDA 200G	25,61403465271%
BRILHANTE EM PO 1 KG , PRODUTOS DIVERSOS BALANCA	161	3,11713457107544%		
			BRILHANTE EM PO 1 KG	25,5555553436279%
BOMBRIL C/8 , PRODUTOS DIVERSOS BALANCA	175	3,38818979263306%		
			BOMBRIL C/8	28,9735107421875%
ESPONJA DE LOUCA SCOTCH-BRITE , PRODUTOS DIVERSOS BALANCA	79	1,52952563762665%		
			ESPONJA DE LOUCA SCOTCH-BRITE	29,044116973877%
ACUCAR CRISTAL ESTRELA , PRODUTOS DIVERSOS BALANCA	115	2,22652459144592%		
			ACUCAR CRISTAL ESTRELA	26,3761463165283%
VINAGRE BRANCO PRINZ , PRODUTOS DIVERSOS BALANCA	96	1,85866403579712%		
			VINAGRE BRANCO PRINZ	28,656717300415%
ARROZ SANTOS T1 , PRODUTOS DIVERSOS BALANCA	88	1,70377540588379%		
			ARROZ SANTOS T1	26,0355033874512%
ALVEJANTE QBOA , PRODUTOS DIVERSOS BALANCA	85	1,64569211006165%		
			ALVEJANTE QBOA	25%
OVOS GRANJA , PRODUTOS DIVERSOS BALANCA	83	1,60696995258331%		
			OVOS GRANJA	31,2030067443848%
REFRIGERANTE PEPSI , PRODUTOS DIVERSOS BALANCA	82	1,58760893344879%		
			REFRIGERANTE PEPSI	31,4176254272461%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
DETERGENTE LIMPOL LIMÃO , PRODUTOS DIVERSOS BALANCA	70	1,35527586936951%		
			DETERGENTE LIMPOL LIMÃO	25%
FEIJAO PRETO FRITZ FRIDA , PRODUTOS DIVERSOS BALANCA	93	1,80058085918427%		
			FEIJAO PRETO FRITZ FRIDA	29,9035377502441%
LIXA D AGUA , OLEO SOJA PRIMOR	205	3,96902227401733%		
			LIXA D AGUA	40,0390625%
CAFE FRITZ FRIDA 200G , OLEO SOJA PRIMOR	121	2,34269118309021%		
			CAFE FRITZ FRIDA 200G	42,4561386108398%
ALVEJANTE QBOA , OLEO SOJA PRIMOR	144	2,78799605369568%		
			ALVEJANTE QBOA	42,3529396057129%
BRILHANTE EM PO 1 KG , OLEO SOJA PRIMOR	250	4,84027099609375%		
			BRILHANTE EM PO 1 KG	39,6825408935547%
			OLEO SOJA PRIMOR	27,5938186645508%
FARINHA DE MILHO LUCIRENE , OLEO SOJA PRIMOR	169	3,27202320098877%		
			FARINHA DE MILHO LUCIRENE	38,6727676391602%
FERMENTO SAFT-INSTANT , OLEO SOJA PRIMOR	152	2,94288468360901%		
			FERMENTO SAFT-INSTANT	45,1038589477539%
BATATA , OLEO SOJA PRIMOR	263	5,09196519851685%		
			BATATA	37,1468925476074%
			OLEO SOJA PRIMOR	29,0286979675293%
BANANA , OLEO SOJA PRIMOR	132	2,55566310882568%		
			BANANA	26,3473052978516%
ERVA AMAVEL , OLEO SOJA PRIMOR	304	5,88576936721802%		
			ERVA AMAVEL	35,764705657959%
			OLEO SOJA PRIMOR	33,554084777832%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
CEBOLA , OLEO SOJA PRIMOR	229	4,43368816375732%		
			CEBOLA	42,2509231567383%
			OLEO SOJA PRIMOR	25,2759380340576%
BOMBRIL C/8 , OLEO SOJA PRIMOR	248	4,80154895782471%		
			BOMBRIL C/8	41,0596008300781%
			OLEO SOJA PRIMOR	27,373067855835%
REFRIGERANTE PEPSI , OLEO SOJA PRIMOR	76	1,4714423418045%		
			REFRIGERANTE PEPSI	29,1187744140625%
FERMENTO SAFT 500G , OLEO SOJA PRIMOR	98	1,89738631248474%		
			FERMENTO SAFT 500G	33,447097783203%
ESPONJA DE LOUCA SCOTCH-BRITE , OLEO SOJA PRIMOR	90	1,74249756336212%		
			ESPONJA DE LOUCA SCOTCH-BRITE	33,0882339477539%
NESCAU ACTIGEN-E , OLEO SOJA PRIMOR	159	3,07841229438782%		
			NESCAU ACTIGEN-E	42,1750679016113%
COUVE CHINESA , OLEO SOJA PRIMOR	111	2,14908027648926%		
			COUVE CHINESA	41,2639389038086%
ERVA MATE AMIZADE , OLEO SOJA PRIMOR	134	2,59438538551331%		
			ERVA MATE AMIZADE	34,7150268554688%
MILHO PARA PIPOCA FRIDA , OLEO SOJA PRIMOR	157	3,03969025611877%		
			MILHO PARA PIPOCA FRIDA	38,9578170776367%
SAGU FRITZ FRIDA , OLEO SOJA PRIMOR	125	2,42013549804688%		
			SAGU FRITZ FRIDA	41,8060188293457%
ARROZ SANTOS T1 , OLEO SOJA PRIMOR	141	2,72991275787354%		
			ARROZ SANTOS T1	41,7159767150879%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
FERMENTO QUIMICO ROYAL , OLEO SOJA PRIMOR	136	2,63310742378235%		
			FERMENTO QUIMICO ROYAL	42,9022064208984%
TOMATE , OLEO SOJA PRIMOR	152	2,94288468360901%		
			TOMATE	30,4609222412109%
SAL SUL , OLEO SOJA PRIMOR	202	3,91093897819519%		
			SAL SUL	45,4954948425293%
ACUCAR CRISTAL ESTRELA , OLEO SOJA PRIMOR	184	3,56243944168091%		
			ACUCAR CRISTAL ESTRELA	42,2018356323242%
VINAGRE BRANCO PRINZ , OLEO SOJA PRIMOR	155	3,00096797943115%		
			VINAGRE BRANCO PRINZ	46,2686576843262%
LEITE CONDENSADO TRADICONAL , OLEO SOJA PRIMOR	89	1,7231365442276%		
			LEITE CONDENSADO TRADICONAL	32,4817504882813%
OVOS GRANJA , OLEO SOJA PRIMOR	75	1,45208132266998%		
			OVOS GRANJA	28,1954879760742%
MARGARINA SOYA , OLEO SOJA PRIMOR	102	1,97483062744141%		
			MARGARINA SOYA	34,6938781738281%
DETERGENTE LIMPOL LIMAO , OLEO SOJA PRIMOR	104	2,01355266571045%		
			DETERGENTE LIMPOL LIMAO	37,1428565979004%
FEIJAO PRETO FRITZ FRIDA , OLEO SOJA PRIMOR	152	2,94288468360901%		
			FEIJAO PRETO FRITZ FRIDA	48,8745994567871%
FARINHA DE MILHO LUCIRENE , ERVA AMAVEL	179	3,46563410758972%		
			FARINHA DE MILHO LUCIRENE	40,9610977172852%
FERMENTO QUIMICO ROYAL , ERVA AMAVEL	90	1,74249756336212%		
			FERMENTO QUIMICO ROYAL	28,3911666870117%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
ALVEJANTE QBOA , ERVA AMAVEL	110	2,12971925735474%		
			ALVEJANTE QBOA	32,3529396057129%
CAFE FRITZ FRIDA 200G , ERVA AMAVEL	109	2,11035823822021%		
			CAFE FRITZ FRIDA 200G	38,2456130981445%
CEBOLA , ERVA AMAVEL	165	3,1945788860321%		
			CEBOLA	30,4428043365479%
BATATA , ERVA AMAVEL	192	3,71732807159424%		
			BATATA	27,1186447143555%
BOMBRIL C/8 , ERVA AMAVEL	208	4,0271053314209%		
			BOMBRIL C/8	34,437084197998%
FERMENTO SAFT-INSTANT , ERVA AMAVEL	140	2,71055173873901%		
			FERMENTO SAFT-INSTANT	41,543025970459%
BRILHANTE EM PO 1 KG , ERVA AMAVEL	214	4,14327192306519%		
			BRILHANTE EM PO 1 KG	33,9682540893555%
			ERVA AMAVEL	25,1764698028564%
FEIJAO PRETO FRITZ FRIDA , ERVA AMAVEL	121	2,34269118309021%		
			FEIJAO PRETO FRITZ FRIDA	38,9067535400391%
FERMENTO SAFT 500G , ERVA AMAVEL	91	1,76185870170593%		
			FERMENTO SAFT 500G	31,0580196380615%
VINAGRE BRANCO PRINZ , ERVA AMAVEL	124	2,40077447891235%		
			VINAGRE BRANCO PRINZ	37,0149269104004%
ESPONJA DE LOUCA SCOTCH-BRITE , ERVA AMAVEL	95	1,8393030166626%		
			ESPONJA DE LOUCA SCOTCH-BRITE	34,9264717102051%

BI

N

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
LIXA D AGUA , ERVA AMAVEL	186	3,60116171836853%		
			LIXA D AGUA	36,328125%
NESCAU ACTIGEN-E , ERVA AMAVEL	107	2,07163596153259%		
			NESCAU ACTIGEN-E	28,3819637298584%
COUVE CHINESA , ERVA AMAVEL	104	2,01355266571045%		
			COUVE CHINESA	38,6617088317871%
MILHO PARA PIPOCA FRIDA , ERVA AMAVEL	140	2,71055173873901%		
			MILHO PARA PIPOCA FRIDA	34,7394523620605%
SAGU FRITZ FRIDA , ERVA AMAVEL	102	1,97483062744141%		
			SAGU FRITZ FRIDA	34,113712310791%
SAL SUL , ERVA AMAVEL	175	3,38818979263306%		
			SAL SUL	39,4144134521484%
ACUCAR CRISTAL ESTRELA , ERVA AMAVEL	168	3,25266218185425%		
			ACUCAR CRISTAL ESTRELA	38,5321083068848%
ARROZ SANTOS T1 , ERVA AMAVEL	121	2,34269118309021%		
			ARROZ SANTOS T1	35,7988166809082%
MARGARINA SOYA , ERVA AMAVEL	87	1,68441438674927%		
			MARGARINA SOYA	29,5918369293213%
DETERGENTE LIMPOL LIMA0 , ERVA AMAVEL	83	1,60696995258331%		
			DETERGENTE LIMPOL LIMA0	29,6428565979004%
LIXA D AGUA , BATATA	129	2,49757981300354%		
			LIXA D AGUA	25,1953125%
ALVEJANTE QBOA , BATATA	96	1,85866403579712%		
			ALVEJANTE QBOA	28,235294342041%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
BRILHANTE EM PO 1 KG , BATATA	161	3,11713457107544%		
			BRILHANTE EM PO 1 KG	25,5555553436279%
FARINHA DE MILHO LUCIRENE , BATATA	111	2,14908027648926%		
			FARINHA DE MILHO LUCIRENE	25,4004573822021%
FERMENTO SAFT-INSTANT , BATATA	115	2,22652459144592%		
			FERMENTO SAFT-INSTANT	34,1246299743652%
BANANA , BATATA	163	3,15585660934448%		
			BANANA	32,5349311828613%
CEBOLA , BATATA	275	5,32429838180542%		
			CEBOLA	50,7380065917969%
			BATATA	38,8418083190918%
MACA , BATATA	91	1,76185870170593%		
			MACA	30,4347820281982%
VINAGRE BRANCO PRINZ , BATATA	92	1,78121972084045%		
			VINAGRE BRANCO PRINZ	27,4626865386963%
COUVE CHINESA , BATATA	112	2,16844153404236%		
			COUVE CHINESA	41,6356887817383%
BANANA PRATA , BATATA	99	1,91674733161926%		
			BANANA PRATA	31,3291130065918%
ARRDZ SANTOS T1 , BATATA	110	2,12971925735474%		
NESCAU ACTIGEN-E , BATATA	96	1,85866403579712%		
			NESCAU ACTIGEN-E	25,4641914367676%
TOMATE , BATATA	200	3,87221693992615%		
			TOMATE	40,0801620483398%
			BATATA	28,2485866546631%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
OVOS GRANJA , BATATA	87	1,68441438674927%		
			OVOS GRANJA	32,7067680358887%
SAL SUL , BATATA	140	2,71055173873901%		
			SAL SUL	31,5315322875977%
ACUCAR CRISTAL ESTRELA , BATATA	119	2,30396890640259%		
			ACUCAR CRISTAL ESTRELA	27,2935771942139%
MILHO PARA PIPOCA FRIDA , BATATA	111	2,14908027648926%		
			MILHO PARA PIPOCA FRIDA	27,5434246063232%
FERMENTO QUIMICO ROYAL , BATATA	81	1,56824779510498%		
			FERMENTO QUIMICO ROYAL	25,5520496368408%
FEIJAO PRETO FRITZ FRIDA , BATATA	88	1,70377540588379%		
			FEIJAO PRETO FRITZ FRIDA	28,2958202362061%
LEITE CONDENSADO TRADICONAL , BATATA	69	1,33591485023499%		
			LEITE CONDENSADO TRADICONAL	25,1824817657471%
DETERGENTE LIMPOL LIMA0 , BATATA	73	1,41335916519165%		
			DETERGENTE LIMPOL LIMA0	26,0714282989502%
LIXA D AGUA , BRILHANTE EM PO 1 KG	185	3,58180069923401%		
			LIXA D AGUA	36,1328125%
			BRILHANTE EM PO 1 KG	29,3650798797607%
CAFE FRITZ FRIDA 200G , BRILHANTE EM PO 1 KG	89	1,7231365442276%		
			CAFE FRITZ FRIDA 200G	31,2280693054199%
ALVEJANTE QBOA , BRILHANTE EM PO 1 KG	138	2,67182970046997%		
			ALVEJANTE QBOA	40,5882339477539%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
FARINHA DE MILHO LUCIRENE , BRILHANTE EM PO 1 KG	152	2,94288468360901%		
			FARINHA DE MILHO LUCIRENE	34,7826080322266%
FERMENTO SAFT-INSTANT , BRILHANTE EM PO 1 KG	110	2,12971925735474%		
			FERMENTO SAFT-INSTANT	32,6409492492676%
FERMENTO SAFT 500G , BRILHANTE EM PO 1 KG	96	1,85866403579712%		
			FERMENTO SAFT 500G	32,7645034790039%
ESPONJA DE LOUCA SCOTCH-BRITE , BRILHANTE EM PO 1 KG	96	1,85866403579712%		
			ESPONJA DE LOUCA SCOTCH-BRITE	35,294116973877%
MILHO PARA PIPOCA FRIDA , BRILHANTE EM PO 1 KG	116	2,24588584899902%		
			MILHO PARA PIPOCA FRIDA	28,7841186523438%
COUVE CHINESA , BRILHANTE EM PO 1 KG	80	1,54888677597046%		
			COUVE CHINESA	29,7397766113281%
BOMBRIL C/8 , BRILHANTE EM PO 1 KG	201	3,89157795906067%		
			BOMBRIL C/8	33,2781448364258%
			BRILHANTE EM PO 1 KG	31,9047622680664%
CEBOLA , BRILHANTE EM PO 1 KG	143	2,76863503456116%		
			CEBOLA	26,3837642669678%
NESCAU ACTIGEN-E , BRILHANTE EM PO 1 KG	110	2,12971925735474%		
			NESCAU ACTIGEN-E	29,1777191162109%
SAGU FRITZ FRIDA , BRILHANTE EM PO 1 KG	77	1,49080348014832%		
			SAGU FRITZ FRIDA	25,7525081634521%
SAL SUL , BRILHANTE EM PO 1 KG	133	2,57502412796021%		
			SAL SUL	29,9549541473389%
FERMENTO QUIMICO ROYAL , BRILHANTE EM PO 1 KG	82	1,58760893344879%		
			FERMENTO QUIMICO ROYAL	25,8675079345703%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
VINAGRE BRANCO PRINZ , BRILHANTE EM PO 1 KG	106	2,05227494239807%		
			VINAGRE BRANCO PRINZ	31,6417903900146%
DETERGENTE LIMPOL LIMAO , BRILHANTE EM PO 1 KG	91	1,76185870170593%		
			DETERGENTE LIMPOL LIMAO	32,5%
MARGARINA SOYA , BRILHANTE EM PO 1 KG	81	1,56824779510498%		
			MARGARINA SOYA	27,5510196685791%
FEIJAO PRETO FRITZ FRIDA , BRILHANTE EM PO 1 KG	81	1,56824779510498%		
			FEIJAO PRETO FRITZ FRIDA	26,045015335083%
CEBOLA , BOMBRIL C/8	145	2,80735731124878%		
			CEBOLA	26,7527675628662%
SAGU FRITZ FRIDA , BOMBRIL C/8	98	1,89738631248474%		
			SAGU FRITZ FRIDA	32,7759208679199%
ESPONJA DE LOUCA SCOTCH-BRITE , BOMBRIL C/8	113	2,18780255317688%		
			ESPONJA DE LOUCA SCOTCH-BRITE	41,544116973877%
LIXA D AGUA , BOMBRIL C/8	218	4,22071647644043%		
			LIXA D AGUA	42,578125%
			BOMBRIL C/8	36,092716217041%
NESCAU ACTIGEN-E , BOMBRIL C/8	109	2,11035823822021%		
			NESCAU ACTIGEN-E	28,9124660491943%
FERMENTO SAFT 500G , BOMBRIL C/8	108	2,09099698066711%		
			FERMENTO SAFT 500G	36,8600692749023%
CAFE FRITZ FRIDA 200G , BOMBRIL C/8	96	1,85866403579712%		
			CAFE FRITZ FRIDA 200G	33,684211730957%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
FARINHA DE MILHO LUCIRENE , BOMBRIL C/8	146	2,8267183303833%		
			FARINHA DE MILHO LUCIRENE	33,409610748291%
MILHO PARA PIPOCA FRIDA , BOMBRIL C/8	128	2,47821879386902%		
			MILHO PARA PIPOCA FRIDA	31,7617874145508%
SAL SUL , BOMBRIL C/8	156	3,02032923698425%		
			SAL SUL	35,1351356506348%
			BOMBRIL C/8	25,8278141021729%
ALVEJANTE QBOA , BOMBRIL C/8	124	2,40077447891235%		
			ALVEJANTE QBOA	36,470588684082%
ARROZ SANTOS T1 , BOMBRIL C/8	94	1,81994187831879%		
			ARROZ SANTOS T1	27,8106517791748%
ACUCAR CRISTAL ESTRELA , BOMBRIL C/8	115	2,22652459144592%		
			ACUCAR CRISTAL ESTRELA	26,3761463165283%
VINAGRE BRANCO PRINZ , BOMBRIL C/8	115	2,22652459144592%		
			VINAGRE BRANCO PRINZ	34,3283576965332%
FERMENTO QUIMICO ROYAL , BOMBRIL C/8	87	1,68441438674927%		
			FERMENTO QUIMICO ROYAL	27,4447956085205%
FERMENTO SAFT-INSTANT , BOMBRIL C/8	89	1,7231365442276%		
			FERMENTO SAFT-INSTANT	26,409496307373%
FEIJAO PRETO FRITZ FRIDA , BOMBRIL C/8	107	2,07163596153259%		
			FEIJAO PRETO FRITZ FRIDA	34,405143737793%
MARGARINA SOYA , BOMBRIL C/8	95	1,8393030166626%		
			MARGARINA SOYA	32,3129234313965%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
DETERGENTE LIMPOL LIMA0 , BOMBRIL C/8	88	1,70377540588379%		
			DETERGENTE LIMPOL LIMA0	31,4285717010498%
BANANA , CEBOLA	139	2,69119071960449%		
			BANANA	27,7445106506348%
			CEBOLA	25,6457557678223%
VINAGRE BRANCO PRINZ , CEBOLA	90	1,74249756336212%		
			VINAGRE BRANCO PRINZ	26,8656711578369%
NESCAU ACTIGEN-E , CEBOLA	104	2,01355266571045%		
			NESCAU ACTIGEN-E	27,5862064361572%
COUVE CHINESA , CEBOLA	94	1,81994187831879%		
			COUVE CHINESA	34,9442367553711%
BANANA PRATA , CEBOLA	80	1,54888677597046%		
			BANANA PRATA	25,3164558410645%
ARROZ SANTOS T1 , CEBOLA	85	1,64569211006165%		
			ARROZ SANTOS T1	25,147928237915%
CAFE FRITZ FRIDA 200G , CEBOLA	78	1,51016461849213%		
			CAFE FRITZ FRIDA 200G	27,3684215545654%
MILHO PARA PIPOCA FRIDA , CEBOLA	108	2,09099698066711%		
			MILHO PARA PIPOCA FRIDA	26,7990074157715%
SAL SUL , CEBOLA	122	2,36205220222473%		
			SAL SUL	27,4774780273438%
TOMATE , CEBOLA	197	3,814133644104%		
			TOMATE	39,4789581298828%
			CEBOLA	36,3468627929688%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
OVOS GRANJA , CEBOLA	81	1,56824779510498%		
			OVOS GRANJA	30,4511280059814%
FERMENTO SAFT-INSTANT , CEBOLA	93	1,80058085918427%		
			FERMENTO SAFT-INSTANT	27,5964393615723%
FEIJAO PRETO FRITZ FRIDA , CEBOLA	87	1,68441438674927%		
			FEIJAO PRETO FRITZ FRIDA	27,9742755889893%
CAFE FRITZ FRIDA 200G , LIXA D AGUA	79	1,52952563762665%		
			CAFE FRITZ FRIDA 200G	27,7192974090576%
ALVEJANTE QBOA , LIXA D AGUA	94	1,81994187831879%		
			ALVEJANTE QBOA	27,6470584869385%
FARINHA DE MILHO LUCIRENE , LIXA D AGUA	131	2,53630208969116%		
			FARINHA DE MILHO LUCIRENE	29,9771175384521%
			LIXA D AGUA	25,5859375%
FERMENTO SAFT-INSTANT , LIXA D AGUA	85	1,64569211006165%		
			FERMENTO SAFT-INSTANT	25,2225513458252%
ESPONJA DE LOUCA SCOTCH-BRITE , LIXA D AGUA	76	1,4714423418045%		
			ESPONJA DE LOUCA SCOTCH-BRITE	27,9411773681641%
FERMENTO SAFT 500G , LIXA D AGUA	105	2,03291392326355%		
			FERMENTO SAFT 500G	35,8361778259277%
SAGU FRITZ FRIDA , LIXA D AGUA	81	1,56824779510498%		
			SAGU FRITZ FRIDA	27,0903015136719%
MILHO PARA PIPOCA FRIDA , LIXA D AGUA	118	2,28460788726807%		
			MILHO PARA PIPOCA FRIDA	29,2803974151611%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
VINAGRE BRANCO PRINZ , LIXA D AGUA	97	1,87802517414093%		
			VINAGRE BRANCO PRINZ	28,9552230834961%
MARGARINA SOYA , LIXA D AGUA	79	1,52952563762665%		
			MARGARINA SOYA	26,8707485198975%
MACA , BANANA	101	1,9554694890976%		
			MACA	33,7792625427246%
BANANA PRATA , TOMATE	89	1,7231365442276%		
			BANANA PRATA	28,1645565032959%
MACA , TOMATE	82	1,58760893344879%		
			MACA	27,4247493743896%
FERMENTO SAFT-INSTANT , SAL SUL	98	1,89738631248474%		
			FERMENTO SAFT-INSTANT	29,0801181793213%
ARROZ SANTOS T1 , SAL SUL	88	1,70377540588379%		
			ARROZ SANTOS T1	26,0355033874512%
ALVEJANTE QBOA , SAL SUL	89	1,7231365442276%		
			ALVEJANTE QBOA	26,1764698028564%
VINAGRE BRANCO PRINZ , SAL SUL	87	1,68441438674927%		
			VINAGRE BRANCO PRINZ	25,9701499938965%
FEIJAO PRETO FRITZ FRIDA , SAL SUL	84	1,62633109092712%		
			FEIJAO PRETO FRITZ FRIDA	27,0096454620361%
FERMENTO SAFT 500G , FARINHA DE MILHO LUCIRENE	77	1,49080348014832%		
			FERMENTO SAFT 500G	26,2798633575439%
FEIJAO PRETO FRITZ FRIDA , FARINHA DE MILHO LUCIRENE	79	1,52952563762665%		
			FEIJAO PRETO FRITZ FRIDA	25,4019298553467%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
FERMENTO SAFT-INSTANT , ACUCAR CRISTAL ESTRELA	85	1,64569211006165%		
			FERMENTO SAFT-INSTANT	25,2225513458252%
ARROZ SANTOS T1 , ACUCAR CRISTAL ESTRELA	99	1,91674733161926%		
			ARROZ SANTOS T1	29,2899398803711%
FEIJAO PRETO FRITZ FRIDA , ACUCAR CRISTAL ESTRELA	91	1,76185870170593%		
			FEIJAO PRETO FRITZ FRIDA	29,2604503631592%
SAGU FRITZ FRIDA , MILHO PARA PIPOCA FRIDA	94	1,81994187831879%		
			SAGU FRITZ FRIDA	31,4381275177002%
VINAGRE BRANCO PRINZ , MILHO PARA PIPOCA FRIDA	84	1,62633109092712%		
			VINAGRE BRANCO PRINZ	25,0746269226074%
ESPONJA DE LOUCA SCOTCH-BRITE , MILHO PARA PIPOCA FRIDA	69	1,33591485023499%		
			ESPONJA DE LOUCA SCOTCH-BRITE	25,3676471710205%
MACA , BANANA PRATA	97	1,87802517414093%		
			MACA	32,4414710998535%
			BANANA PRATA	30,6962032318115%
CEBOLA , ERVA AMAVEL , OLEO SOJA PRIMOR	91	1,76185870170593%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	29,9342098236084%
BATATA , ERVA AMAVEL , OLEO SOJA PRIMOR	103	1,99419164657593%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	33,8815803527832%
BOMBRILO C/8 , ERVA AMAVEL , OLEO SOJA PRIMOR	104	2,01355266571045%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	34,2105255126953%
BRILHANTE EM PO 1 KG , ERVA AMAVEL , OLEO SOJA PRIMOR	104	2,01355266571045%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	34,2105255126953%
LIXA D AGUA , ERVA AMAVEL , OLEO SOJA PRIMOR	92	1,78121972084045%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	30,2631587982178%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
SAL SUL , ERVA AMAVEL , OLEO SOJA PRIMOR	84	1,62633109092712%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	27,6315784454346%
ACUCAR CRISTAL ESTRELA , ERVA AMAVEL , OLEO SOJA PRIMOR	80	1,54888677597046%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	26,3157901763916%
PRODUTOS DIVERSOS BALANCA , ERVA AMAVEL , OLEO SOJA PRIMOR	79	1,52952563762665%		
			ERVA AMAVEL , OLEO SOJA PRIMOR	25,9868412017822%
BANANA , CEBOLA , BATATA	75	1,45208132266998%		
			CEBOLA , BATATA	27,2727279663086%
ERVA AMAVEL , CEBOLA , BATATA	85	1,64569211006165%		
			CEBOLA , BATATA	30,9090900421143%
BOMBRILO C/8 , CEBOLA , BATATA	78	1,51016461849213%		
			CEBOLA , BATATA	28,3636360168457%
OLEO SOJA PRIMOR , CEBOLA , BATATA	133	2,57502412796021%		
			CEBOLA , BATATA	48,3636360168457%
BRILHANTE EM PO 1 KG , CEBOLA , BATATA	85	1,64569211006165%		
			CEBOLA , BATATA	30,9090900421143%
TOMATE , CEBOLA , BATATA	111	2,14908027648926%		
			CEBOLA , BATATA	40,3636360168457%
LIXA D AGUA , BATATA , OLEO SOJA PRIMOR	73	1,41335916519165%		
			BATATA , OLEO SOJA PRIMOR	27,7566547393799%
BRILHANTE EM PO 1 KG , BATATA , OLEO SOJA PRIMOR	92	1,78121972084045%		
			BATATA , OLEO SOJA PRIMOR	34,9809875488281%
FERMENTO SAFT-INSTANT , BATATA , OLEO SOJA PRIMOR	68	1,31655371189117%		
			BATATA , OLEO SOJA PRIMOR	25,8555126190186%
BOMBRILO C/8 , BATATA , OLEO SOJA PRIMOR	74	1,43272018432617%		
			BATATA , OLEO SOJA PRIMOR	28,1368827819824%

Item de Suporte	Soma	Suporte	Item de Confiança	Confiança
NESCAU ACTIGEN-E , BATATA , OLEO SOJA PRIMOR	67	1,29719269275665%		
			BATATA , OLEO SOJA PRIMOR	25,475284576416%
TOMATE , BATATA , OLEO SOJA PRIMOR	93	1,80058085918427%		
			BATATA , OLEO SOJA PRIMOR	35,3612174987793%
SAL SUL , BATATA , OLEO SOJA PRIMOR	84	1,62633109092712%		
			BATATA , OLEO SOJA PRIMOR	31,9391632080078%

ANEXO G – Código de Implementação do formulário principal

```
unit formprinc;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, IBQuery, DB, IBCustomDataSet, IBTable, IBDatabase,
  Buttons, ExtCtrls, ComCtrls;

type
  TForm1 = class(TForm)
    Button6: TButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton2: TSpeedButton;
    procedure Button6Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);

  private
    { Private declarations }

  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  totalRegistros : Integer;

implementation

uses formDataModule, formProdescription, formResult, formRegisterCompras,
  formDataMining;
{$R *.dfm}

procedure TForm1.Button6Click(Sender: TObject);
begin
  frmCadProdutos.show;
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
  frmResult.show;
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
  frmRegisterCompras.show;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
  frmDataMining.show;
end;
```

ANEXO H – Código de Implementação do formulário de “Registro de Compras”

```

unit formRegisterCompras;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls;

type
  TfrmRegisterCompras = class(TForm)
    openFile1: TOpenDialog;
    Button1: TButton;
    Button3: TButton;
    Memo1: TMemo;
    Memo2: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Edit2: TEdit;
    Label8: TLabel;
    BitBtn2: TBitBtn;
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    SaveDialog1: TSaveDialog;
    BitBtn1: TBitBtn;
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmRegisterCompras: TfrmRegisterCompras;

implementation

uses formDataModule;

{$R *.dfm}

procedure TfrmRegisterCompras.Button1Click(Sender: TObject);
begin
  if ( Edit1.Text <> '' ) then
    begin
      if ( Edit2.Text <> '' ) then
        begin
          if ( Memo2.Lines.GetText <> '' ) then
            begin
              Memo1.Lines.Add('Selecionando arquivo de dados....');
            end;
          end;
        end;
      end;
    end;
end;

```

```

        if ( openFile1.Execute ) then
        begin
            Memo2.Lines.LoadFromFile(openFile1.FileName);

            Memo1.Lines.Add('Arquivo Selecionado.');
```

end;

```

            Button3.Click;
        end
    else
        showMessage('Por favor, repita a operação pois o arquivo seleciona está inválido.');
```

end

```

    else
        showMessage('Por favor, indique uma coluna relativa ao CÓDIGO DO PRODUTO
ADQUIRIDO.');
```

end

```

    else
        showMessage('Por favor, indique uma coluna relativa ao IDENTIFICADOR DA COMPRA.');
```

end;

```

procedure TfrmRegisterCompras.Button3Click(Sender: TObject);
var
    textoArquivoPreProcessamento : String; //Guardará o texto que vier do arquivo para o Memo1 ao
    abrir o arquivo

    i : Integer; //Iterador do arquivo de texto

    dado : String; //Variável auxiliar para registrar o dado em si, para posteriormente
    trabalhá-lo

    iNumItemFromFile : Integer; //Iterador do número do item (coluna) do arquivo;

    keyValueItem : String; //Registra a chave de referência a um valor de determinado item
    valueItem : String; //Registra um valor de determinado item

    iNumKeyValueItem : Integer; //Regitra o número do item(coluna) que referencia o
    keyValueItem - este valor é uma chave
    iNumValueItem : Integer; //Regitra o número do item(coluna) que referencia o valueItem -
    este valor é uma chave

    sqlInsert : String; //Auxiliar para registra a instrução de insert do sql;
    sqlValues : String; //Auxiliar para registra os VALUES de uma instrução sql;

    iTotValueTotItem : Integer; //Contador que armazenará o total de valores para cada keyItem
begin
    iNumItemFromFile := 1;

    iNumKeyValueItem := StrToInt(Edit1.Text);
    iNumValueItem := StrToInt(Edit2.Text);

    keyValueItem := '';
    valueItem := '';

    { **** 12/10/2007 14:45
      #13, equivale a ENTER no Delphi, ou no nosso exemplo a quebra de linha
    }

    { **** 12/10/2007 14:45
      Deletar todos os registros da tabela de compras, que irá registrar
      as compras em si; e da tabela tmpValues, que irá ser uma tabela
      temporária auxiliar para o registro das compras
    }

```

```
dmDataModule.tbCompras.Open;
dmDataModule.qryTbCompras.SQL.Text := 'DELETE FROM compras';
dmDataModule.qryTbCompras.ExecSQL;
```

```
dmDataModule.tbTmpValues.Open;
dmDataModule.qryTbTmpValues.SQL.Text := 'DELETE FROM tmpValues';
dmDataModule.qryTbTmpValues.ExecSQL;
```

```
{ **** 12/10/2007 14:45
  Amazenado na variável textoArquivoPreProcessamento, o conteúdo
  de um Memo (Memo2) que foi preenchido na lida do arquivo na
  seleção do mesmo em "Registrar Compras"
}
textoArquivoPreProcessamento := '';
textoArquivoPreProcessamento := Memo2.Lines.GetText;

Memo1.Lines.Add('Registrando no Banco de Dados ... ');

{ **** 12/10/2007 14:45
  Percorre-se o arquivo lido e os dados são inseridos na tabela auxiliar (tmpvalues), para
  posteriormente serem lidos desta tabela para a transposição dos mesmo no processo
  de de registro das compras na tabela de compras.
}
i := 1;

while i <= (Length(textoArquivoPreProcessamento)) do
begin
  // Impressão dos logs se marcado o radioButton no formulário.
  if ( RadioButton1.Checked = true ) then
    Memo1.Lines.Add('Linha ' + IntToStr(i) + ' de ' +
IntToStr(Length(textoArquivoPreProcessamento)) + ' registrada. ');

  /*** [4] Verifica se o ultimo caracter é quebra de linha - ENTER **
  if ( (textoArquivoPreProcessamento[i] = #13) ) then
  begin
    //Encontrou um ENTER = fim do registro atual

    //Verifica se a coluna atual é referente ao keyValueltem ou ao valueltem
    if ( iNumItemFromFile = iNumKeyValueltem ) then
    begin
      //Coluna é relativa a chave primária
      keyValueltem := dado;
    end
    else if ( iNumItemFromFile = iNumValueltem ) then
      //Coluna é relativa ao valor
      valueltem := dado;

    if ( valueltem = '' ) then
    begin
      //Valor está em branco na base de dados. Dirty Date
      if ( RadioButton1.Checked = true ) then
        Memo1.Lines.Add('Compra registrada e não informado o produto. Identificador de
Compra = ' + keyValueltem);
      end
      else if ( keyValueltem = '' ) then
      begin
        //Chave primária está em branco
        if ( RadioButton1.Checked = true ) then
          Memo1.Lines.Add('Existe IDENTIFICADOR DE COMPRA em branco na linha ' +
IntToStr(i) + ' do arquivo. ');
        end
      end
    end
  end
```

```

else
begin
  //Dados estão informados e ok.

  dmDataModule.qryTbTmpValues.SQL.Clear;
  dmDataModule.qryTbTmpValues.SQL.Add('SELECT keyValueItem, valueItem FROM
tmpValues WHERE keyValueItem = ' + keyValueItem + ' AND valueItem = ' + "" + valueItem + "");
  dmDataModule.qryTbTmpValues.Prepare;
  dmDataModule.qryTbTmpValues.Open;

  if ( ( VarToStr(dmDataModule.qryTbTmpValues.FieldValues['keyValueItem']) <> " ) ) then
  begin
    //Já existe um cadastro para esta nota fiscal e este produto. Isto é um ERRO
    if ( RadioButton1.Checked = true ) then
      Memo1.Lines.Add('O produto ' + valueItem + ' está duplicado na compra de
IDENTIFICADOR DE COMPRA = ' + keyValueItem);
    end
  else
  begin
    sqlInsert := "";
    sqlInsert := 'INSERT INTO tmpValues(keyValueItem, valueItem) VALUES ( ' +
keyValueItem + ', ' + "" + valueItem + "" )';

    dmDataModule.qryTbTmpValues.SQL.Clear;
    dmDataModule.qryTbTmpValues.SQL.Text := sqlInsert;
    dmDataModule.qryTbTmpValues.ExecSQL;
  end;
end;

//Reinicializa variáveis
dado := "";
iNumItemFromFile := 1;
keyValueItem := "";
valueItem := "";

//Incrementa-se 1 em i, senão, ele considerará o enter
//como caracter para o dado posterior
i := i + 1;
end
else
begin
  //Iteração na linha
  if ( (textoArquivoPreProcessamento[i] = ',') ) then
  begin
    //Verifica se a coluna atual é referente ao keyValueItem ou ao valueItem
    if ( iNumItemFromFile = iNumKeyValueItem ) then
    begin
      keyValueItem := dado;
    end
    else if ( iNumItemFromFile = iNumValueItem ) then
    begin
      valueItem := dado;
    end;

    iNumItemFromFile := iNumItemFromFile + 1;

    dado := "";
  end
  else
  begin
    //Ainda está no dado
    //Verifica se não é aspas dulas, daí considera

```

```

        if ( textoArquivoPreProcessamento[i] <> "" ) then
        begin
            //Adiciona caracter por caracter em dado, formando o código do produto
            dado := dado + textoArquivoPreProcessamento[i];
        end;
    end;

end;

i := i + 1;
end;

//Fechar tabela de compras, tmpValues registrando as alterações
dmDataModule.tbCompras.Close;
dmDataModule.tbTmpValues.Close;

{ **** 12/10/2007 14:45
  Seleciona-se todos os itens registrados na tabela tmpValues para trabalhalos
  de forma a inserir corretamente na tabela de compras;

  Detalhe que em tbTmpValues utiliza-se o comando DISTINCT para ter-se todos os
  keyValueItem registrados, posteriormente, selecionará-se para cada um destes
  os valores para os inserts correspondentes utilizando keyValueItem - ValueItem
}
dmDataModule.tbTmpValues.Open;
dmDataModule.qryTbTmpValues.SQL.Text := 'SELECT DISTINCT keyValueItem FROM
tmpValues';
dmDataModule.qryTbTmpValues.Prepare;
dmDataModule.qryTbTmpValues.Open;

//Devido a maneira de trabalho do Delphi, é criada mais de uma instância para a mesma
//tabela para facilitar o trabalho;
dmDataModule.tbTmpValues2.Open;
dmDataModule.tbTmpValues3.Open;

while ( not dmDataModule.qryTbTmpValues.Eof ) do
begin
    sqlInsert := "";
    sqlValues := "";
    iTotValueToItem := 1; //Esta variável tem um papel muito importante, no sentido que controlará
    o produtoId<NUMERO>
        //que está sendo inserido, evitando duplicações.

    dmDataModule.qryTbTmpValues2.SQL.Text := 'SELECT valueItem FROM tmpValues WHERE
keyValueItem = ' + VarToStr(dmDataModule.qryTbTmpValues.FieldValues['keyValueItem']) ;
    dmDataModule.qryTbTmpValues2.Prepare;
    dmDataModule.qryTbTmpValues2.Open;

    sqlInsert := ' INSERT INTO compras (notaFiscalId ';
    sqlValues := ' VALUES ( ' +
    VarToStr(dmDataModule.qryTbTmpValues.FieldValues['keyValueItem']);

    while ( not dmDataModule.qryTbTmpValues2.Eof ) do
    begin
        if ( VarToStr(dmDataModule.qryTbTmpValues2.FieldValues['valueItem']) <> "" ) then
        begin
            //Existe valor. Se o valor for em branco não é inserido, visando a limpeza dos dados
            sqlInsert := sqlInsert + ', produtoId' + IntToStr(iTotValueToItem);
            sqlValues := sqlValues + ', ' +
            VarToStr(dmDataModule.qryTbTmpValues2.FieldValues['valueItem'] + "" ) ;

            iTotValueToItem := iTotValueToItem + 1;
        end;
    end;
end;

```

```
end;
//else Problema nos dados. Não deve inserir. Dado em branco na tabela temporária

dmDataModule.qryTbTmpValues2.Next;
end;

sqlInsert := sqlInsert + ' ) ';
sqlValues := sqlValues + ' ) ';

sqlInsert := sqlInsert + sqlValues;

dmDataModule.qryTbTmpValues3.SQL.Text := sqlInsert;
dmDataModule.qryTbTmpValues3.ExecSQL;

dmDataModule.qryTbTmpValues.Next;
end;

dmDataModule.tbTmpValues2.Close;
dmDataModule.tbTmpValues3.Close;

Memo1.Lines.Add('Registrado no Banco de Dados. ');
end;

procedure TfrmRegisterCompras.BitBtn1Click(Sender: TObject);
begin
    SaveDialog1.Execute;
    Memo1.Lines.SaveToFile(SaveDialog1.FileName);
end;

end.
```

ANEXO I – Código de Implementação do formulário de “Cadastrar Produtos”

```
unit formProdescription;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons;

type
  TfrmCadProdutos = class(TForm)
    Memo1: TMemo;
    openFile1: TOpenDialog;
    Button1: TButton;
    Button2: TButton;
    Memo2: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label6: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    RadioGroup1: TRadioGroup;
    RadioButton2: TRadioButton;
    RadioButton1: TRadioButton;
    BitBtn1: TBitBtn;
    SaveDialog1: TSaveDialog;
    BitBtn2: TBitBtn;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmCadProdutos: TfrmCadProdutos;

implementation

uses formDataModule;

{$R *.dfm}

procedure TfrmCadProdutos.Button1Click(Sender: TObject);
begin
  if ( Edit1.Text <> '' ) then
    begin
      if ( Edit2.Text <> '' ) then
        begin
          if ( Memo1.Lines.GetText <> '' ) then
            begin
              if ( openFile1.Execute ) then
                begin
```

```

Memo1.Clear;
Memo1.Lines.LoadFromFile(openFile1.FileName);
Memo2.Lines.Add('Arquivo Selecionado.');
```

```

    Button2.Click;
end;
end
else
    showMessage('Por favor, repita a operação pois o arquivo seleciona está inválido.');
```

```

end
else
    showMessage('Por favor, indique uma coluna relativa a DESCRIÇÃO DO PRODUTO.');
```

```

end
else
    showMessage('Por favor, indique uma coluna relativa ao CÓDIGO DO PRODUTO.');
```

```

end;
```

```

procedure TfrmCadProdutos.Button2Click(Sender: TObject);
var
    textoArquivoPreProcessamento : String; //Guardará o texto que vier do arquivo para o Memo1 ao
    abrir o arquivo

    i : Integer; //Iterador do arquivo de texto

    dado : String; //Variável auxiliar para registrar o dado em si, para posteriormente
    trabalhá-lo

    iNumItemFromFile : Integer; //Iterador do número do item (coluna) do arquivo;

    produtold : String; //Registro a chave de referência a um valor de determinado item
    proDescription : String; //Registro um valor de determinado item

    iNumKeyValueltem : Integer; //Regitra o número do item(coluna) que referencia o
    keyValueltem - este valor é uma chave
    iNumValueltem : Integer; //Regitra o número do item(coluna) que referencia o valueltem -
    este valor é uma chave

    sqlInsert : String;
begin
    iNumItemFromFile := 1;

    iNumKeyValueltem := StrToInt(Edit1.Text);
    iNumValueltem := StrToInt(Edit2.Text);

    produtold := '';
    proDescription := '';

    //13, equivale a ENTER no Delphi, ou no nosso exemplo a quebra de linha
    { **** 14/10/2007 23:46
      Deletar todos os registros anteriores na tabela proDescription.
      Esta tabela irá armazenar as referências para os produtos - descrição
    }
    dmDataModule.tbProdescription.Open;
    dmDataModule.qryTbProdescription.SQL.Text := 'DELETE FROM proDescription';
    dmDataModule.qryTbProdescription.ExecSQL;

    textoArquivoPreProcessamento := '';
    textoArquivoPreProcessamento := Memo1.Lines.GetText;

    { **** 14/10/2007 23:46

```

```

    Percorre-se todo o arquivo salvo na variável textoArquivoPreProcessamento
}
i := 1;

Memo2.Lines.Add('*** Iniciando cadastro de produtos *** ');

while i <= (Length(textoArquivoPreProcessamento)) do
begin
    // Impressão dos logs se marcado o radioButton no formulário.
    if ( RadioButton1.Checked = true ) then
        Memo2.Lines.Add('Linha ' + IntToStr(i) + ' de ' +
IntToStr(Length(textoArquivoPreProcessamento)) + ' registrada. ');

    /** [4] Verifica se o ultimo caracter é quebra de linha - ENTER **
    if ( (textoArquivoPreProcessamento[i] = #13) ) then
    begin
        //Encontrou um ENTER = fim do registro atual

        //Verifica se a coluna atual é referente ao produtold ou ao valueltem
        if ( iNumItemFromFile = iNumKeyValueltem ) then
        begin
            //Coluna é relativa a chave primária
            produtold := dado;
        end
        else if ( iNumItemFromFile = iNumValueltem ) then
            //Coluna é relativa ao valor
            proDescription := dado;

        if ( proDescription = '' ) then
        begin
            if ( RadioButton1.Checked = true ) then
                Memo2.Lines.Add('Descrição em branco para este produto. Código do Produto = ' +
produtold);
            end
            else if ( produtold = '' ) then
            begin
                if ( RadioButton1.Checked = true ) then
                    Memo2.Lines.Add('Existe CÓDIGO DO PRODUTO em branco na linha ' + IntToStr(i) +
do arquivo. ');
                end
                else
                begin
                    //Dados estão informados e ok.
                    dmDataModule.qryTbProdescription.SQL.Clear;
                    dmDataModule.qryTbProdescription.SQL.Add('SELECT produtold, description FROM
proDescription WHERE produtold = ' + '' + produtold + '' + ' AND description = ' + '' + proDescription
+ '');
                    dmDataModule.qryTbProdescription.Prepare;
                    dmDataModule.qryTbProdescription.Open;

                    if ( ( VarToStr(dmDataModule.qryTbProdescription.FieldValues['produtold']) <> '' ) ) then
                    begin
                        if ( RadioButton1.Checked = true ) then
                            Memo2.Lines.Add('O descrição ' + proDescription + ' está duplicada para o produto de
CÓDIGO DE PRODUTO = ' + produtold);
                        end
                        else
                        begin
                            sqlInsert := '';
                            sqlInsert := 'INSERT INTO proDescription(produtold, description) VALUES ( ' + '' +
produtold + '' + ', ' + '' + proDescription + '' + ')';

```

```

dmDataModule.qryTbProdescription.SQL.Clear;
dmDataModule.qryTbProdescription.SQL.Text := sqlInsert;
dmDataModule.qryTbProdescription.ExecSQL;
end;
end;

//Reinicializa variáveis
dado := "";
iNumItemFromFile := 1;
produtold := "";
proDescription := "";

//Incrementa-se 1 em i, senã, ele considerará o enter
//como caracter para o dado posterior
i := i + 1;
end
else
begin
//Iteração na linha
if ( (textoArquivoPreProcessamento[i] = ',') ) then
begin
//Verifica se a coluna atual é referente ao produtold ou ao proDescription
if ( iNumItemFromFile = iNumKeyValueltem ) then
begin
produtold := dado;
end
else if ( iNumItemFromFile = iNumValueltem ) then
begin
proDescription := dado;
end;

iNumItemFromFile := iNumItemFromFile + 1;

dado := "";
end
else
begin
//Ainda está no dado
//Verifica se não é aspas dulas, daí considera
if ( textoArquivoPreProcessamento[i] <> "" ) then
begin
dado := dado + textoArquivoPreProcessamento[i];
end;
end;

end;

end;

i := i + 1;
end;

Memo2.Lines.Add('*** Produtos cadastrados. *** ');

dmDataModule.tbProdescription.Close;
end;

procedure TfrmCadProdutos.BitBtn1Click(Sender: TObject);
begin
SaveDialog1.Execute;
Memo2.lines.SaveToFile(SaveDialog1.FileName);
end;

end.

```

ANEXO J – Código de Implementação do formulário de “Mineração de Dados”

```
unit formDataMining;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, IBQuery, DB, IBCustomDataSet, IBTable, IBDatabase,
  Buttons, ExtCtrls, ComCtrls;

type
  TfrmDataMining = class(TForm)
    Button7: TButton;
    SpeedButton1: TSpeedButton;
    Button5: TButton;
    Label1: TLabel;
    Label3: TLabel;
    edtSuport: TEdit;
    itemFrequente: TEdit;
    Memo2: TMemo;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    RadioGroup1: TRadioGroup;
    RadioButton2: TRadioButton;
    RadioButton1: TRadioButton;
    SaveDialog1: TSaveDialog;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    Label2: TLabel;
    procedure Button7Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmDataMining: TfrmDataMining;
  totalRegistros : Integer;

implementation

uses formDataModule;

{$R *.dfm}

procedure TfrmDataMining.Button7Click(Sender: TObject);
begin
  if ( (itemFrequente.Text <> "") OR (edtSuport.Text <> "") ) then
    begin
      if ( (itemFrequente.Text <> "") AND (edtSuport.Text <> "") ) then
        begin
          showMessage('Como você informou um ITEM FREQUENTE e um SUPORTE MÍNIMO, o item
que será utilizado pelo algoritmo será o SUPORTE MÍNIMO informado.');
          end;
        end;
    end;
end;
```

```

        SpeedButton1.Click;

        Button5.Click;
    end
    else
    begin
        showMessage('Por favor, informe um ITEM FREQUENTE ou um SUPORTE MÍNIMO.');
```

end;

```

end;

procedure TfrmDataMining.SpeedButton1Click(Sender: TObject);
var
    Teste      : Variant;

    qtdColumnsTable : Integer; //Quantidade de colunas da tabela principal (a gerar valores) neste
    exemplo, é tabela "Compras"

    itColumnsTable : Integer;

    auxKeyControlId, auxSoma : Variant;

    auxFloat : real;

    countRegisterCompra : integer;
begin
    DecimalSeparator := '.';

    Memo2.Lines.Add('Gerando 1 itemSet ... ');

    totalRegistros := 0;

    dmDataModule.tbCompras.Open;
    dmDataModule.tbProdutosId.Open;

    { **** 05/11/2007
      Deletar os registros de produtosId já cadastrados para
      gerar novos
    }
    dmDataModule.qryTbProdutosId.SQL.Clear;
    dmDataModule.qryTbProdutosId.SQL.Add('DELETE FROM produtosid');
    dmDataModule.qryTbProdutosId.Open;

    //dmDataModule.tbCompras.Open;
    dmDataModule.qryTbCompras.SQL.Clear;
    dmDataModule.qryTbCompras.SQL.Add('SELECT * FROM compras');
    dmDataModule.qryTbCompras.Prepare;
    dmDataModule.qryTbCompras.Open;

    { **** 05/11/2007
      Selecciona-se todos os registros da tabela de compras para serem analisados.
      Para cada ocorrência de produto em uma determinada compra, será gerado um item
      na tabela de produtosId. O algoritmo verificará se o produto já existe cadastrado,
      se já existir, é feito o update com a soma incrementada de um, senão o insert;
    }
    if ( dmDataModule.qryTbCompras.Prepared ) then
    begin
        qtdColumnsTable := dmDataModule.qryTbCompras.FieldCount - 1; // Menos 1 aqui, devido a
        coluna de chave primária
    end
end;

```



```

totalRegistros := totalRegistros + 1;

dmDataModule.qryTbCompras.Next;
end;

{ **** 05/11/2007
  Cálculo do suporte para os itens
}
dmDataModule.qryTbProdutosId.SQL.Clear;
dmDataModule.qryTbProdutosId.SQL.Add('SELECT keyControllId, produtos, soma FROM
produtosId WHERE quantProdutos = 1');
dmDataModule.qryTbProdutosId.Prepare;
dmDataModule.qryTbProdutosId.Open;

while ( not dmDataModule.qryTbProdutosId.Eof ) do
begin
  //Suporte = num trans entre A e B / Total de transações

  //auxFloat := dmDataModule.qryTbProdutosId.FieldValues['soma'];

  auxFloat := ( StrToFloat(VarToStr(dmDataModule.qryTbProdutosId.FieldValues['soma'])) *
100) / totalRegistros; //(auxFloat / totalRegistros) * 100 ;

  dmDataModule.qryTbProdutosId2.SQL.Clear;
  dmDataModule.qryTbProdutosId2.SQL.Add('UPDATE produtosId SET suporte = ' + '''' +
FloatToStr(auxFloat) + '''' + ' WHERE keyControllId = ' + '''' +
VarToStr(dmDataModule.qryTbProdutosId.FieldValues['keyControllId']));
  dmDataModule.qryTbProdutosId2.ExecSQL;

  dmDataModule.qryTbProdutosId.Next;
end;
end;

dmDataModule.tbCompras.Close;
dmDataModule.tbProdutosId.Close;

Memo2.Lines.Add('Gerado 1 itemSet. ');
end;

/** Função que retorna a quantidade de colunas de compras
Function quantityColumnsFromCompras() : integer;
begin
  dmDataModule.qryTbCompras.SQL.Clear;
  dmDataModule.qryTbCompras.SQL.Add('SELECT * FROM compras');
  dmDataModule.qryTbCompras.Prepare;
  dmDataModule.qryTbCompras.Open;

  //Select e if para descobrir a quantidade de colunas da tabela
  if ( dmDataModule.qryTbCompras.Prepared ) then
    //Menos 1 acima devido a coluna de chave primária ALTERAR futuramente para ler as todas as
    colunas da tabela
    Result := (dmDataModule.qryTbCompras.FieldCount - 1);
  end;
end;

/** Retorna um produtoId utilizando keyControllId
Function getProdutoUsingKeyControllId(keyControllId : integer) : TIBQuery;
begin
  dmDataModule.qryTbProdutosId2.SQL.Clear;
  dmDataModule.qryTbProdutosId2.SQL.Add('SELECT * FROM produtosId WHERE keyControllId = '
+ IntToStr(keyControllId));

```

```
dmDataModule.qryTbProdutosId2.Prepare;
dmDataModule.qryTbProdutosId2.Open;
```

```
Result := dmDataModule.qryTbProdutosId2;
end;
```

```
procedure TfrmDataMining.Button5Click(Sender: TObject);
```

```
var
```

```
  qtdColsTable : Integer;      //Quantidade de colunas que serão lidas do arquivo que contém
  os dados;
```

```
  itColsTable : Integer;      //Iterador das colunas do arquivo que contém os dados;
```

```
  currentItemSet : Integer;   //ItemSet atual, ou seja, itemSet que está corrente na análise;
```

```
  totalItemSet : Integer;     //Total de itemSets que serão analisados
```

```
  Teste : Variant;           //Variável utilizada como auxiliar. Mudar nome posteriormente
```

```
  somaAux : Integer;         //Variável que auxiliar para guardar a soma de ocorrências de um
  keyValue em um ItemSet
```

```
  confAux : Integer;        //Variável que auxiliar para guardar a confiança de itens
```

```
  keyControlIdAux : Integer; //Variável utilizada para guardar o keyControlIdAux nas inserções
  e updates da tabela
```

```
  whereSql : String;        //Variável auxiliar que guarda as cláusulas where para os selects na
  tabela de compras
```

```
  currentProdutold : Variant; //Armazena produtold atual do laço de repetição
```

```
  currentKeyControlld : Variant; //Armazena keyControlld atual do laço de repetição
```

```
  currentKeyControlldAux : Variant; //Armazena keyControlldAux atual do laço de repetição
```

```
  currentProdutoldAux : Variant; //Variável auxiliar que armazena o produtold. Ver utilidade
  posteriormente
```

```
  axKeyControlld1 : Variant; //Auxiliar para keyControlld
```

```
  currentProdutoldFromCompra : Variant; //Produtold de compra
```

```
  produtosNOTSelect : array of string; //Array com produtos que estão na análise do itemSet e
  portando não devem fazer parte
```

```
  //da seleção e inserção de dados pra evitar incoerencia dos dados
```

```
  itProdutosNOTSelect : integer; //Iterador do array de produtos que não são da seleção
```

```
  isProdutoldInNotSelect : boolean; //Flag de controle
```

```
  auxFloat : Variant;
```

```
  axKeyControlldAux2 : Variant;
```

```
  axKeyControlld2 : Variant;
```

```
  axProdutos2 : Variant;
```

```
  axProdutosArray2 : array of string;
```

```
  itAxProdutosArray2 : integer;
```

```
  axKeyControlWhile2 : Variant;
```

```
  axI2, axJ2, axZ2 : Integer;
```

```
  auxContArray : Integer;
```

```
  axIsUpdate2 : boolean;
```

```
  axFlagFor2 : boolean;
```

```
  axCountFlag2: integer;
```

```

prodsIdItemSetAnterior : TIBQuery; // Produtos do item set anterior;
prodObjectAux          : TIBQuery; // Objeto de produto

prodObjectAux2        : TIBQuery;

prodObjectAux3        : TIBQuery;

axFlagFor3 : boolean;

keyControllIdToUpdate : Variant;

keyControllIdArray : array of string; //Neste array, colocarei os keyControllId que serão inseridos
//só que estes keyControllIds serão keyControllIdAux pois terei
//que verificar se o array de produtos que quero inserir, sem o
//currentProdutold, já não está neste array (os produtos )

//vou ter que comparar os dados deste array com o que está
//em produtosNOTSelect. Considerando o currentProdutold.
itKeyControllIdArray : integer;

auxKeyControllIdArray : array of string;
itAuxKeyControllIdArray : integer;

justAnalised : boolean;
axCountFlag3 : integer;

itemSetControl : Variant;

suporteAux : Real;

begin
  DecimalSeparator := '.';

  dmDataModule.tbCompras.Open;
  dmDataModule.tbProdutosId.Open;

  dmDataModule.qryTbProdutosIdConfianc.SQL.Clear;
  dmDataModule.qryTbProdutosIdConfianc.SQL.Add('DELETE FROM produtosidconfianc;');
  dmDataModule.qryTbProdutosIdConfianc.ExecSQL;

  qtdColumnsTable := quantityColumnsFromCompras(); //Total de colunas da tabela compras menos
a coluna da chave primária;

  { **** 05/11/2007
    Laço para percorrer todos os itemSets cadastrados
  }

  itemSetControl := true;

  currentItemSet := 2;

  while ( VarToStr(itemSetControl) <> '' ) do
    begin
      Memo2.Lines.Add('Gerando ' + IntToStr(currentItemSet) + ' itemSet ... ');

      dmDataModule.qryTbProdutosId.SQL.Clear;
      dmDataModule.qryTbProdutosId.SQL.Add('DELETE FROM produtosid WHERE quantProdutos =
' + IntToStr(currentItemSet));
      dmDataModule.qryTbProdutosId.Open;

      { **** 10/11/2007 09:24

```

Seleciono todos os produtos de um itemSet anterior, com os critérios de item frequente (soma maior que) este item frequente, pois estes produtos serão utilizados posteriormente para cruzar eles com os outros que estão registrados na tabela de compras. Ou seja, percorrerá-se linha por linha onde existe estes produtos.

```

}
if ( edtsuport.text <> " ) then
begin
  dmDataModule.qryTbProdutosId.SQL.Clear;
  dmDataModule.qryTbProdutosId.SQL.Add('SELECT * FROM produtosId WHERE
quantProdutos = ' + IntToStr(currentItemSet-1) + ' AND suporte >= ' + "" + edtSuport.Text + "" +
'ORDER BY soma DESC' );
  dmDataModule.qryTbProdutosId.Prepare;
  dmDataModule.qryTbProdutosId.Open;
end
else
begin
  dmDataModule.qryTbProdutosId.SQL.Clear;
  dmDataModule.qryTbProdutosId.SQL.Add('SELECT * FROM produtosId WHERE
quantProdutos = ' + IntToStr(currentItemSet-1) + ' AND soma >= ' + itemFrequente.Text + ' ORDER
BY soma DESC ');
  dmDataModule.qryTbProdutosId.Prepare;
  dmDataModule.qryTbProdutosId.Open;
end;

itemSetControl := dmDataModule.qryTbProdutosId.FieldValues['keyControllId']; //Só para
armazenar um valor, para dizer que pode continuar com o laço dos itensSets

keyControllIdArray := nil;

SetLength(keyControllIdArray, 1000); // Este tamanho é para os registro que irão ser inseridos,
ou seja,
// não tenho esse número. Estou utilindo 1000, mas futuramente, pode
// ter uma técnica para calcular este valor
itKeyControllIdArray := 0;

while ( not dmDataModule.qryTbProdutosId.Eof ) do //not dmDataModule.qryTbProdutosId.Eof )
do
begin
  currentProdutold := dmDataModule.qryTbProdutosId.FieldValues['produtos'];
  currentKeyControllId := dmDataModule.qryTbProdutosId.FieldValues['keyControllId'];
  currentKeyControllIdAux := dmDataModule.qryTbProdutosId.FieldValues['keyControllIdAux'];

  memo2.Lines.Add('Analisando ' + VarToStr(currentProdutold) + ' soma = ' +
VarToStr(dmDataModule.qryTbProdutosId.FieldValues['soma']));

  if ( RadioButton1.Checked = true ) then
    memo2.Lines.add('##### currentProdutold -> ' + VArToStr(currentProdutold));

  { **** 10/11/2007 09:24
    Cria a cláusula WHERE correta que será utilizada para selecionar todas as compras
    que possuem o produto atual (currentProdutold) que é o produto corrente deste while
    referente aos produtos do itemSet anterior
  }
  whereSql := ' ( ' ;
  axKeyControllId1 := dmDataModule.qryTbProdutosId.FieldValues['keyControllId'];

  produtosNOTSelect := nil; //Array que irá registrar todos os produtos do itemSet anterior + o
produto que

```

```

//quere-se inserir. Será utilizado posteriormente para verificação de
registros que
//devem serem inseridos ou que dessem serem atualizados incrementando a
soma;
itProdutosNOTSelect := 0;
SetLength(produtosNOTSelect, currentItemSet); // Seta o tamanho do array;

//Tentativa de percorrer até chegar em um null para keyControlldAux, onde será o primeiro
inserido para o itemSet
while ( axKeyControlld1 <> null ) do
begin
//Objeto com os dados de produto de keyControlld registrado em axKeyControlld1
prodObjectAux := getProdutoUsingKeyControlld(StrToInt(VarToStr(axKeyControlld1)));

currentProdutoldAux := prodObjectAux.FieldValues['produtos'];

produtosNOTSelect[itProdutosNOTSelect] := currentProdutoldAux;
itProdutosNOTSelect := itProdutosNOTSelect + 1;

//Percorrerá todas as colunas de compras para formar o where correto do sql
for itColsTable := 1 to qtdColsTable do
begin
if ( itColsTable = 1 ) then
begin
whereSql := whereSql + ' produtold' + IntToStr(itColsTable) + ' = ' + '' +
VarToStr(currentProdutoldAux) + ''';
end
else
begin
whereSql := whereSql + ' OR produtold' + IntToStr(itColsTable) + ' = ' + '' +
VarToStr(currentProdutoldAux) + ''';
end;
end;

axKeyControlld1 := prodObjectAux.FieldValues['keyControlldAux'];

if ( axKeyControlld1 = null ) then
begin
whereSql := whereSql + ' )';
end
else
begin
whereSql := whereSql + ' ) AND ( ';
end;
end;

{ **** 10/11/2007 09:24
  Aplica o where, selecionando em compras o produto atual do laço referente a todos os
  produtos do itemSet anterior que satisfazem o itemFrequente - suporte mínimo
}
dmDataModule.qryTbCompras.SQL.Clear;
dmDataModule.qryTbCompras.SQL.Add('SELECT * FROM compras WHERE ' + whereSql);
dmDataModule.qryTbCompras.Prepare;
dmDataModule.qryTbCompras.Open;

//Verifica se existe ao menos uma compra registrada para este critério passado no WHERE
Teste := dmDataModule.qryTbCompras.FieldValues['notafiscalld'];

if ( VarToStr(Teste) <> '' ) then
begin
while ( not dmDataModule.qryTbCompras.Eof ) do
begin

```

```

produtosNOTSelect[currentItemSet-1] := "";

for itColumnsTable := 1 to qtdColumnsTable do
begin
    currentProdutoldFromCompra :=
dmDataModule.qryTbCompras.FieldValues['produtold' + IntToStr(itColumnsTable)];

    //Desconsidera as colunas nulas em compras;
    if ( VarToStr(currentProdutoldFromCompra) <> " ) then
    begin
        { **** 12/11/2007 14:56
            Verificará se este produto na compra para o produto do itemSet anterior
selecionado,
            satisfaz a condição do itemFrequente - suporte mínimo cadastrado no seu
cadastro como
            produto único (1 itemSet de produtosId)
        }
dmDataModule.qryTbProdutosId3.SQL.Clear;

        if ( edtsuport.text <> " ) then
        begin
            dmDataModule.qryTbProdutosId3.SQL.Add('SELECT * FROM produtosId
WHERE quantProdutos = 1 AND suporte >= ' + "" + edtSuport.Text + "" + ' AND produtos = ' + "" +
VarToStr(currentProdutoldFromCompra) + "" ');
            end
            else
            begin
                dmDataModule.qryTbProdutosId3.SQL.Add('SELECT * FROM produtosId
WHERE quantProdutos = 1 AND soma >= ' + itemFrequente.Text + ' AND produtos = ' + "" +
VarToStr(currentProdutoldFromCompra) + "" ');
                end;

            dmDataModule.qryTbProdutosId3.Prepare;
            dmDataModule.qryTbProdutosId3.Open;

            Teste := dmDataModule.qryTbProdutosId3.FieldValues['keyControllId'];

            {
                **** 10/11/2007 11:47
                If para desconsiderar o produto, se este tem menos que o itemFrequente citado
no cadastro geral de produtos (cadastro para 1 itemSet). Hoje, compara com o
itemFrequente
                de apenas 1 (quantidade de produtos). Talvez o ideal seja, no futuro, ver para
qual keyControllId
                se está inserindo e comparar com este keyControllId.
            }
            if ( VarToStr(Teste) <> " ) then
            begin
                { **** 10/11/2007 11:54
                    Agora, temos que pegar todos os produtos deste keyControllId e nosso produto
atual
                    - currentProdutoldFromCompra, colocar em um array, e verificar se esta
sequencia
                    de produtos já não está cadastrada no itemSet ao qual queremos inserir.

                    O keyControllId anterior, que será o keyControllIdAux na inserção futura, e será
a
                    chave para pesquisarmos os produtos anteriores está armazenado em
CURRENTKEYCONTROLID

                    O produto o qual iremos verificar, e posteriormente inserir se tudo der certo
está

```

na variável CURRENTPRODUTOIDFROMCOMPRA;

Em produtosNOTSelect, já tenho armazenado os produtos que pertencem ao keyControllId atual do array, dentro deste for, sempre o currentProdutoldFromCompra, que é o que eu quero inserir posteriormente se der tudo certo.

```

}
{
**** 15/11/2007 16:32
    Esta flag irá determinar, se for FALSE, que o elemento que quero inserir já
    existe
    no array de elementos - ProdutosNOTselect, ou seja, iria ser duplicado. Se
    o elemento não existir
    pode proceguir
}
axFlagFor3 := true;

for axI2 := 0 to (length(produtosNOTSelect) - 1) do
begin
    if (produtosNOTSelect[axI2] = VarToStr(currentProdutoldFromCompra)) then
    begin
        axFlagFor3 := false;
        break;
    end;
end;

if ( axFlagFor3 = true ) then
begin
    //Insiro no array meu produto atual, quem vem do for. Tem o -1 pois a
    primeira posição deste array é 0;
    produtosNOTSelect[currentItemSet-1] :=
    VarToStr(currentProdutoldFromCompra);

    { **** 13/11/2007
      Retorna todos os produtos já cadastrados para o itemSet atual.
    }
    dmDataModule.qryTbProdutosId4.SQL.Clear;
    dmDataModule.qryTbProdutosId4.SQL.Add('SELECT * FROM produtosId
WHERE quantProdutos = ' + IntToStr(currentItemSet));
    dmDataModule.qryTbProdutosId4.Prepare;
    dmDataModule.qryTbProdutosId4.Open;

    //Verifica se já existem produtos para o itemSet que quero inserir
    if ( VarToStr(dmDataModule.qryTbProdutosId4.FieldValues['keyControllId'])
<> " ) then
begin
    {
    **** 15/11/2007 22:18
      Esta fla será utilizada para marcar se é um update
      senão, será um insert do item;
    }
    axIsUpdate2 := false;
    keyControllIdToUpdate := "";

    while ( (not dmDataModule.qryTbProdutosId4.Eof) AND (axIsUpdate2 =
false) ) do
begin
    {

```

**** 10/11/2007 14:18

Variável que irá ser o iterador do array que guardará os produtos já cadastrados para este itemSet. Irá ser modificado em cada iteração, incluindo os produtos do itemSet da vez do while;

O iterador para este array será ITAXPRODUTOSARRAY2;
O array dos produtos mesmo será AXPProdutosArray2;

```
}
axProdutosArray2 := nil;
itAxProdutosArray2 := 0;
```

SetLength(axProdutosArray2, currentItemSet); // Seta o tamanho do array; -1 tinha antes

```
axProdutosArray2[itAxProdutosArray2] :=
dmDataModule.qryTbProdutosId4.FieldValues['produtos'];
itAxProdutosArray2 := itAxProdutosArray2 + 1;
```

```
axKeyControlWhile2 :=
dmDataModule.qryTbProdutosId4.FieldValues['keyControlIdAux'];
```

```
if ( VarToStr(axKeyControlWhile2) <> '' ) then
begin
while ( axKeyControlWhile2 <> null ) do //terá que percorrer todos os
produtos, indo descendo através de keyControlAux
```

```
begin
dmDataModule.qryTbProdutosId6.SQL.Clear;
dmDataModule.qryTbProdutosId6.SQL.Add('SELECT * FROM
produtosId WHERE keyControlId = ' + VarToStr(axKeyControlWhile2));
dmDataModule.qryTbProdutosId6.Prepare;
dmDataModule.qryTbProdutosId6.Open;
```

```
axProdutosArray2[itAxProdutosArray2] :=
dmDataModule.qryTbProdutosId6.FieldValues['produtos'];
itAxProdutosArray2 := itAxProdutosArray2 + 1;
```

```
axKeyControlWhile2 :=
dmDataModule.qryTbProdutosId6.FieldValues['keyControlIdAux'];
end;
end;
```

```
{
**** 10/11/2207
```

Agora, tenho que fazer as comparações entre os 2 arrays.

Se os elementos de ambos os arrays coexistirem, então deve ser feito o UPDATE do registro de keyControlId que deve ser somado de 1;

Se algum dos elementos for diferente, ou seja, os dados que quero inserir não existem ainda

na tabela, então é só inserir.

```
}
axCountFlag2 := 0;
```

```
for axI2 := 0 to (length(produtosNOTSelect) - 1) do
begin
for axJ2 := 0 to (length(axProdutosArray2) - 1) do
begin
if ( axProdutosArray2[axJ2] = produtosNOTSelect[axI2] ) then
begin
axCountFlag2 := axCountFlag2 + 1;
break;
end;
```

```

        end;
    end;

    if ( length(produtosNOTSelect) = axCountFlag2 ) then
    begin
        //UPDATE
        axsUpdate2 := true;
        keyControllIdToUpdate :=
dmDataModule.qryTbProdutosId4.FieldValues['keyControllId'];
        end;
        //else //begin //Continuar no laço, se não sair antes de terminar, é
INSERT //end;

        dmDataModule.qryTbProdutosId4.Next;
    end;

    justAnalised := false;

    for axI2 := 0 to (length(keyControllIdArray) - 1) do
    begin
        if ( (keyControllIdArray[axI2] <> "") AND (justAnalised = false) ) then
        begin
            axCountFlag3 := 0;

            dmDataModule.qryTbProdutosId8.SQL.Clear;
            dmDataModule.qryTbProdutosId8.SQL.Add('SELECT * FROM
produtosId WHERE keyControllId = ' + keyControllIdArray[axI2]);
            dmDataModule.qryTbProdutosId8.Prepare;
            dmDataModule.qryTbProdutosId8.Open;

            axKeyControlWhile2 := "";
            axKeyControlWhile2 :=
dmDataModule.qryTbProdutosId8.FieldValues['keyControllIdAux']; //Variável auxiliar já utilizada em
outros laços anteriores

            itAuxKeyControllIdArray := 0;
            auxKeyControllIdArray := nil;
            SetLength(auxKeyControllIdArray, (currentItemSet-1));

            auxKeyControllIdArray[itAuxKeyControllIdArray] :=
dmDataModule.qryTbProdutosId8.FieldValues['produtos'];
            itAuxKeyControllIdArray := itAuxKeyControllIdArray + 1;

            if ( VarToStr(axKeyControlWhile2) <> "" ) then
            begin
                while ( axKeyControlWhile2 <> null ) do //terá que percorrer todos
os produtos, indo descendo através de keyControlAux
                begin
                    dmDataModule.qryTbProdutosId8.SQL.Clear;
                    dmDataModule.qryTbProdutosId8.SQL.Add('SELECT * FROM
produtosId WHERE keyControllId = ' + VarToStr(axKeyControlWhile2));
                    dmDataModule.qryTbProdutosId8.Prepare;
                    dmDataModule.qryTbProdutosId8.Open;

                    auxKeyControllIdArray[itAuxKeyControllIdArray] :=
dmDataModule.qryTbProdutosId8.FieldValues['produtos'];
                    itAuxKeyControllIdArray := itAuxKeyControllIdArray
+ 1;

                    axKeyControlWhile2 :=
dmDataModule.qryTbProdutosId8.FieldValues['keyControllIdAux'];
                    end;
                end;
            end;
        end;
    end;

```

```

for axJ2 := 0 to (length(auxKeyControlldArray) - 1) do
begin
  for axZ2 := 0 to (length(produtosNOTSelect) - 1) do
  begin
    if ( auxKeyControlldArray[axJ2] = produtosNOTSelect[axZ2] )

then
      begin
        //Produto já cadastrado em keyControlld anterior
        axCountFlag3 := axCountFlag3 + 1;
      end;
    end;
  end;

  if ( axCountFlag3 = length(auxKeyControlldArray) ) then
  begin
    justAnalised := true;
    Break;
  end;
end;

end;
end;

if ( justAnalised = false ) then
begin
  if ( axIsUpdate2 = true ) then
  begin
    //Update
    dmDataModule.qryTbProdutosId7.SQL.Clear;
    dmDataModule.qryTbProdutosId7.SQL.Add('SELECT soma FROM
produtosid WHERE keyControlld = ' + VArToStr(keyControlldToUpdate));
    dmDataModule.qryTbProdutosId7.Prepare;
    dmDataModule.qryTbProdutosId7.Open;

    somaAux :=
StrToInt(VarToStr(dmDataModule.qryTbProdutosId7.FieldValues['soma']));
    somaAux := somaAux + 1 ;

    dmDataModule.qryTbProdutosId7.SQL.Clear;
    dmDataModule.qryTbProdutosId7.SQL.Add('UPDATE produtosid SET
soma = ' + IntToStr(somaAux) + ' WHERE keyControlld = ' + VArToStr(keyControlldToUpdate));
    dmDataModule.qryTbProdutosId7.ExecSQL;
  end
  else
  begin
    //Insert
    dmDataModule.qryTbProdutosId7.SQL.Clear;
    dmDataModule.qryTbProdutosId7.SQL.Add('INSERT INTO
produtosId(soma, produtos, quantProdutos, keyControlldAux ) VALUES (1 , ' + '' +
currentProdutoldFromCompra + '' + ' , ' + IntToStr(currentItemSet) + ' , ' +
VarToStr(currentKeyControlld) + ');');
    dmDataModule.qryTbProdutosId7.ExecSQL;
  end;
end;
end
else
begin
  {
  **** 10/11/2007 13:37
  Aqui, irei inserir o primeiro produto do laço, para meu ItemSet.

```

```

que
    Este IF irá verificar se o produto que eu quero inserir, não igual a algum
    já está inserido
    }
    if ( currentProdutold <> currentProdutoldFromCompra ) then
    begin
        dmDataModule.qryTbProdutosId5.SQL.Clear;
        dmDataModule.qryTbProdutosId5.SQL.Add('INSERT INTO
produtosId(soma, produtos, quantProdutos, keyControlldAux)' + ' VALUES (1, ' + '' +
VarToStr(currentProdutoldFromCompra) + '' + ', ' + IntToStr(currentItemSet) + ', ' +
VarToStr(currentKeyControlld) + ') ');
        dmDataModule.qryTbProdutosId5.ExecSQL;
        end;
    end;
    end
    //else //begin //produto igual. Já existe no produtosNOTSelect //end;
    end
    //else //begin //O produto está abaixo do item frequente //end;
    end
    else
    begin
        break;
    end;
    end;

    dmDataModule.qryTbCompras.Next; //Próxima compra retornado no select
    end;
    end;
    //else //begin //Não encontrou dado em pesquisa retornada //end;

    { **** 22/11/2007 11:43
    Aqui irá registrar o keycontrolld analisado do itemSet atual em keyControlldArray
    }
    if ( VarToStr(currentKeyControlld) <> " ) then
    begin
        keyControlldArray[itKeyControlldArray] := VarToStr(currentKeyControlld);
        itKeyControlldArray := itKeyControlldArray + 1;
    end;

    dmDataModule.qryTbProdutosId.Next; //Próximo produto retornado no select
    end;
Memo2.Lines.Add('Gerado ' + IntToStr(currentItemSet) + ' itemSet. ');

    { **** 22/11/2007 11:45
    Geração do suporte e da confiança para os produtos
    }
    memo2.Lines.add('Gerando suporte e grau de confiança para ' + IntToStr(currentItemSet) );
    dmDataModule.qryTbProdutosId.SQL.Clear;
    dmDataModule.qryTbProdutosId.SQL.Add('SELECT * FROM produtosId WHERE quantProdutos
= ' + IntToStr(currentItemSet));
    dmDataModule.qryTbProdutosId.Prepare;
    dmDataModule.qryTbProdutosId.Open;

    while ( not dmDataModule.qryTbProdutosId.Eof ) do
    begin
        //Suporte = num trans entre A e B / Total de transações
        auxFloat := dmDataModule.qryTbProdutosId.FieldValues['soma'];
        somaAux := auxFloat;

        auxFloat := ( somaAux * 100 ) / totalRegistros;

```

```

dmDataModule.qryTbProdutosId2.SQL.Clear;
dmDataModule.qryTbProdutosId2.SQL.Add('UPDATE produtosId SET suporte = ' + "" +
FloatToStr(auxFloat) + "" + ' WHERE keyControllId = ' +
VarToStr(dmDataModule.qryTbProdutosId.FieldValues['keyControllId']));
dmDataModule.qryTbProdutosId2.ExecSQL;

//Confiança = num trans entre A e B / Num trans. conten. A; ou aqui, num trans. conten. B
axKeyControlWhile2 := "";
axKeyControlWhile2 := dmDataModule.qryTbProdutosId.FieldValues['keyControllIdAux'];
//Variável auxiliar já utilizada em outros laços anteriores

keyControllIdToUpdate := dmDataModule.qryTbProdutosId.FieldValues['keyControllId'];
//Variável auxiliar já utilizada em outros laços anteriores
currentProdutold := dmDataModule.qryTbProdutosId.FieldValues['produtos']; //Aqui esta
variável é utilizada para guardar o produto a ser inserido na confiança;

if ( VarToStr(axKeyControlWhile2) <> "" ) then
begin
while ( axKeyControlWhile2 <> null ) do //terá que percorrer todos os produtos, indo
descendo através de keyControlAux
begin
dmDataModule.qryTbProdutosId8.SQL.Clear;
dmDataModule.qryTbProdutosId8.SQL.Add(' SELECT * ' +
' FROM produtosId ' +
' WHERE keyControllId = ' + "" + VarToStr(axKeyControlWhile2)
+ "" );
dmDataModule.qryTbProdutosId8.Prepare;
dmDataModule.qryTbProdutosId8.Open;

confAux := dmDataModule.qryTbProdutosId8.FieldValues['soma'];

auxFloat := ( somaAux / confAux ) * 100;

dmDataModule.qryTbProdutosIdConfianc.SQL.Clear;
dmDataModule.qryTbProdutosIdConfianc.SQL.Add(' INSERT INTO
produtosIdconfianc(keycontrolid, keycontrolidfromprodutos, confianca) VALUES( ' +
VarToStr(keyControllIdToUpdate) + ', ' + VarToStr(axKeyControlWhile2) + ', ' + "" +
FloatToStr(auxFloat) + "" + ');');
dmDataModule.qryTbProdutosIdConfianc.ExecSQL;

axKeyControlWhile2 := dmDataModule.qryTbProdutosId8.FieldValues['keyControllIdAux'];

end;

//Aqui, está implementado para registrar também a confiança para este produtold também;
dmDataModule.qryTbProdutosId8.SQL.Clear;
dmDataModule.qryTbProdutosId8.SQL.Add(' SELECT * ' +
' FROM produtosId ' +
' WHERE produtos = ' + "" + VarToStr(currentProdutold) + "" +
' AND quantProdutos = 1 ; ');
dmDataModule.qryTbProdutosId8.Prepare;
dmDataModule.qryTbProdutosId8.Open;

confAux := dmDataModule.qryTbProdutosId8.FieldValues['soma'];

auxFloat := ( somaAux / confAux ) * 100;

dmDataModule.qryTbProdutosIdConfianc.SQL.Clear;
dmDataModule.qryTbProdutosIdConfianc.SQL.Add(' INSERT INTO
produtosIdconfianc(keycontrolid, keycontrolidfromprodutos, confianca) VALUES( ' +
VarToStr(keyControllIdToUpdate) + ', ' +

```

```

VarToStr(dmDataModule.qryTbProdutosId8.FieldValues['keyControllId']) + ', ' + '' + ' + '
FloatToStr(auxFloat) + '' + ');');
    dmDataModule.qryTbProdutosIdConfianc.ExecSQL;

    axKeyControlWhile2 := dmDataModule.qryTbProdutosId8.FieldValues['keyControllIdAux'];
end;

    dmDataModule.qryTbProdutosId.Next;
end;

    Memo2.Lines.Add('Gerado SUPORTE / CONFIANÇA para ' + IntToStr(currentItemSet) + '
itemSet. ');

    currentItemSet := currentItemSet + 1;
end;

Memo2.Lines.Add('Processo finalizado');

dmDataModule.tbCompras.Close;
dmDataModule.tbProdutosId.Close;
end;

procedure TfrmDataMining.BitBtn1Click(Sender: TObject);
begin
    SaveDialog1.Execute;
    Memo2.Lines.SaveToFile(SaveDialog1.FileName);
end;

end.

```

ANEXO K – Código de Implementação do formulário de “Visualizar Resultados”

```
unit formResult;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, Buttons;

type
  TfrmResult = class(TForm)
    Button1: TButton;
    stringResult: TStringGrid;
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label4: TLabel;
    BitBtn2: TBitBtn;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmResult: TfrmResult;

implementation

uses formDataModule;

{$R *.dfm}

procedure TfrmResult.Button1Click(Sender: TObject);
var
  soma, produtold, produtoDescription, suporte, quantProdutos, keyControlld, oldKeyControlld :
  Variant;

  produtoldFromConfianc, proDescriptionFromConfianc, confianca : Variant;

  itRow : Integer;

  auxKeyControlldAux : Variant;

  keyControlldAux : Variant;

  stringProdutos : string;
begin
  if ( Edit1.Text <> " ) then
    begin
      dmDataModule.tbProdutosId.Open;

      stringResult.Cells[1, 0] := 'Item de Suporte';
      stringResult.Cells[2, 0] := 'Soma';
      stringResult.Cells[3, 0] := 'Suporte';
      stringResult.Cells[4, 0] := 'Item de Confiança';
```

```

stringResult.Cells[5, 0] := 'Confiança';

itRow := 1; //Iterador da linha da grid

{ **** 23/11/2007 19:56
  Selecciona-se todos os produtos, com sua descrição,
  para geração dos resultados
}
dmDataModule.qryTbProdutosIdConfianc.SQL.Clear;
dmDataModule.qryTbProdutosIdConfianc.SQL.Add(' SELECT * FROM produtosIdConfianc A
INNER JOIN produtosId B ON (B.keyControllId = A.keyControllIdFromProdutos) INNER JOIN
proDescription C ON (B.produtos = C.produtold) WHERE A.confianca >= ' + "" + Edit1.Text + "" +
'ORDER BY A.keyControllId, A.confianca DESC; ');
dmDataModule.qryTbProdutosIdConfianc.Prepare;
dmDataModule.qryTbProdutosIdConfianc.Open;

while ( not dmDataModule.qryTbProdutosIdConfianc.Eof ) do
begin
  keyControllId      := dmDataModule.qryTbProdutosIdConfianc.FieldValues['keyControllId'];

  produtoldFromConfianc
dmDataModule.qryTbProdutosIdConfianc.FieldValues['produtold'];
  proDescriptionFromConfianc
dmDataModule.qryTbProdutosIdConfianc.FieldValues['description'];
  confianca          := dmDataModule.qryTbProdutosIdConfianc.FieldValues['confianca'];
  keyControllIdAux
dmDataModule.qryTbProdutosIdConfianc.FieldValues['keyControllIdAux'];

  if ( VarToStr(keyControllId) <> VarToStr(oldKeyControllId) ) then
  begin
    //KeyControllId da tabela de confianças é diferente. Ou seja, novo produto
    dmDataModule.qryTbProdutosId.SQL.Clear;
    dmDataModule.qryTbProdutosId.SQL.Add(' SELECT * FROM produtosId A INNER JOIN
proDescription B ON (A.produtos = B.produtold) WHERE A.keyControllId = ' + VarToStr(keyControllId)
+ ' ORDER BY A.suporte DESC; ');
    dmDataModule.qryTbProdutosId.Prepare;
    dmDataModule.qryTbProdutosId.Open;

    soma          := dmDataModule.qryTbProdutosId.FieldValues['soma'];
    produtold     := dmDataModule.qryTbProdutosId.FieldValues['produtold'];
    produtoDescription := dmDataModule.qryTbProdutosId.FieldValues['description'];
    suporte       := dmDataModule.qryTbProdutosId.FieldValues['suporte'];
    quantProdutos := dmDataModule.qryTbProdutosId.FieldValues['quantProdutos'];

    stringProdutos := VarToStr(produtoDescription);
    auxKeyControllIdAux := dmDataModule.qryTbProdutosId.FieldValues['keyControllIdAux'];

  { **** 23/11/2007 14:56
    Aqui, percorre-se o para os produtos que estão gerando a confiança
  }
  if ( VarToStr(keyControllIdAux) <> "" ) then
  begin
    while ( VarToStr(keyControllIdAux) <> "" ) do
    begin
      dmDataModule.qryTbProdutosId2.SQL.Clear;
      dmDataModule.qryTbProdutosId2.SQL.Add(' SELECT *
      FROM PRODUTOSID A
      INNER JOIN PRODESCRIPTION B
      ON (A.produtos = B.produtold)
      WHERE keyControllId = ' + "" +
      VarToStr(keyControllIdAux) + "" +
      ');
    end
  end
}

```

```

dmDataModule.qryTbProdutosId2.Prepare;
dmDataModule.qryTbProdutosId2.Open;

keyControllIdAux := dmDataModule.qryTbProdutosId2.FieldValues['keyControllIdAux'];

proDescriptionFromConfianc := proDescriptionFromConfianc + ', ' +
VarToStr(dmDataModule.qryTbProdutosId2.FieldValues['description']);
end;
end;

{ **** 23/11/2007 14:56
  Percorrerá os produtos para o keyControllId, ou seja, os que irão gerar o suporte
  na Visualização dos Resultados
}
if ( VarToStr(auxKeyControllIdAux) <> " " ) then
begin
  while ( VarToStr(auxKeyControllIdAux) <> " " ) do
  begin
    dmDataModule.qryTbProdutosId2.SQL.Clear;
    dmDataModule.qryTbProdutosId2.SQL.Add(' SELECT *
      FROM PRODUTOSID A
      INNER JOIN PRODESCRIPTION B
      ON (A.produtos = B.produtold)
      WHERE keyControllId = ' + VarToStr(
VarToStr(auxKeyControllIdAux) + ' ' +
      ');
    dmDataModule.qryTbProdutosId2.Prepare;
    dmDataModule.qryTbProdutosId2.Open;

    auxKeyControllIdAux :=
dmDataModule.qryTbProdutosId2.FieldValues['keyControllIdAux'];

    stringProdutos := stringProdutos +
VarToStr(dmDataModule.qryTbProdutosId2.FieldValues['description']);
    end;
    end;

    stringResult.RowCount := stringResult.RowCount + 1;
    stringResult.Cells[1, itRow] := stringProdutos;
    stringResult.Cells[2, itRow] := VarToStr(soma);
    stringResult.Cells[3, itRow] := VarToStr(suporte) + '%';
    itRow := itRow + 1;
  end
  else
  begin
    //KeyControllId é o mesmo, ou seja, está "caminhando" no keyControllId para confianças
    diferentes
    end;

    stringResult.RowCount := stringResult.RowCount + 1;
    stringResult.Cells[4, itRow] := VarToStr(proDescriptionFromConfianc);
    stringResult.Cells[5, itRow] := VarToStr(confianca) + '%';
    itRow := itRow + 1;

    oldKeyControllId := keyControllId;

    dmDataModule.qryTbProdutosIdConfianc.Next;
  end;
end
else
begin

```

```
        showMessage('Por favor, informe uma CONFIANÇA MÍNIMA para a visualização dos  
resultados.');
```

```
    end;  
end;  
  
end.
```