UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JÚLIO LIMA VIEIRA FILHO

Proposta de Framework para Sistemas de Mineração de Dados Socioeconômicos

São Luís
2013

JÚLIO LIMA VIEIRA FILHO

Proposta de Framework para Sistemas de Mineração de Dados Socioeconômicos

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Geraldo Braz Junior
Mestre em Engenharia de Eletricidade – UFMA

São Luís
2013

Vieira Filho, Júlio Lima

Proposta de Framework para sistemas de mineração de dados socioeconômicos / Júlio Lima Vieira Filho. – São Luís, 2013.

49 f.

Orientador: Geraldo Braz Júnior.
Monografia (Graduação) – Curso de Ciências da Computação da Universidade Federal do Maranhão, 2013.

1. Framework (Desenvolvimento de Software). 2. Análise de dados. 3. Mineração de dado. I. Título.

CDU 004.75

JÚLIO LIMA VIEIRA FILHO

Proposta de Framework para Sistemas de Mineração de Dados Socioeconômicos

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em **12 / 12 / 2013**

BANCA EXAMINADORA

Prof. Msc. Geraldo Braz Junior (Orientador)

Prof. Dr. Anselmo Cardoso de Paiva

Prof.ª Msc. Simara Vieira da Rocha

Aos meus pais Júlio Lima Vieira e Claudete Teles Mendes Vieira.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que sempre esteve à minha frente guiando-me através dos caminhos árduos da vida e me apoiando com Seu amor nos momentos de fraqueza. Aos meus pais, pelo apoio constante e por toda a educação, conselhos, incentivos e repreensões que tornaram meu caráter tão forte quanto o deles. Agradeço também a todos os professores e funcionários do Departamento de Informática, em especial, ao meu orientador Prof. Geraldo Braz Junior por todos os ensinamentos e confiança passados durante a elaboração deste trabalho e ao Prof. Anselmo Cardoso de Paiva pela ajuda e oportunidades oferecidas durante minha graduação. À minha namorada e a todos os meus amigos que torcem pelo meu sucesso assim também como eu, pelo deles.

“Se descubro em mim um desejo que nenhuma experiência deste mundo pode satisfazer, a explicação mais provável é que fui criado para um outro mundo.”

(C. S. Lewis)

RESUMO

Para obtenção de informações com objetivo de analisar a situação social e econômica de uma população é de fundamental importância contar com instrumentos técnicos facilitadores, precisos e de boa qualidade. Os softwares de análise de dados socioeconômicos são amplamente utilizados nesse cenário, pois são capazes de processar quantidades enormes de dados permitindo a obtenção de estatísticas e indicadores socioeconômicos. Esta monografia propõe um framework para a construção de sistemas desse tipo, auxiliando e agilizando o processo de desenvolvimento de softwares de análise de dados socioeconômicos, equipando-os com as funções básicas e gerando novas informações através do processo de mineração de dados.

Palavras-chave: Framework, Análise de dados, Mineração de dados.

ABSTRACT

To analyze social and economic conditions of a population, is important to obtain information through technical resources precisely and with quality. Software capable of analyzing socio-economic data are widely used in this context, due to its capacity to process large amounts of data to obtain statistics and socioeconomic indicators. This work proposes a framework to build socio-economic data analysis software, supporting and accelerating its development process, by integrating basic functions and producing new information through data mining process.

Keywords: Framework, Data Analysis, Data Mining.

SUMÁRIO

LISTA DE FIGURAS

1. INTRODUÇÃO

Um dos objetivos do Governo é identificar os problemas que afetam o desenvolvimento humano e executar políticas públicas a fim de resolvê-los, entretanto para a correta identificação dos problemas se faz necessário um esforço no sentido de coletar informações de natureza socioeconômica da população e então aplicar estudos de análise sobre esses dados para se chegar à fonte do problema que precisa ser resolvido.

Os dados socioeconômicos de uma população se aplicam a diversas áreas como saúde, educação e moradia e vão até temas com renda per capta e índices de mortalidade. O grande número de tipos diferentes de informação e a vasta quantidade de dados coletada pelas instituições de pesquisa e órgãos governamentais fazem da tarefa de análise dos dados, humanamente muito difícil ou impossível.

Dada esta situação os governos e demais instituições desse contexto têm usado sistemas de softwares para o agrupamento e a representação da informação de forma que simplifique sua análise. Estes são sistemas que manipulam dados organizados em séries temporais, baseados na investigação ligada a subclassificações ou localidades, configurando dados multidimensionais.

Esse trabalho tem por objetivo geral propor um framework para o desenvolvimento de sistemas de análise de dados socioeconômicos no sentido de auxiliar a elaboração e a construção de tais sistemas assim como a resolução dos problemas mais comuns encontrados. De forma específica, esse trabalho pretende oferecer meios para o melhor aproveitamento da base de dados de sistemas desse tipo, definindo métodos para a construção de novos dados socioeconômicos a partir dos existentes através de processos de mineração de dados.

Esta monografia está dividida em seis capítulos. O segundo capítulo descreve alguns dos sistemas governamentais mais conhecidos no cenário nacional. No terceiro capítulo são apresentados alguns dos conceitos mais significativos para a compreensão deste trabalho. O Capítulo 4 descreve a proposta do framework expondo suas especificações, modelos e arquitetura. O Capítulo 5 apresenta, como caso de teste, um sistema desenvolvido com o framework. Finalmente, o Capítulo 6 descreve a conclusão do trabalho retratando suas limitações além de sugestões para futuros trabalhos.

2. FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento de software é uma atividade de crescente importância na sociedade contemporânea, pois a utilização de computadores nas mais diversas áreas do conhecimento humano tem gerado uma crescente demanda por soluções computadorizadas.

A aplicação de softwares para a resolução de problemas dos mais variados tipos e temas implica no estudo prévio dos conceitos e métodos relacionados com o problema durante o processo de desenvolvimento do software. Por exemplo, seria improvável a construção de um sistema eficiente de detecção de tumores em imagens radiológicas sem que a equipe de desenvolvimento dominasse o conhecimento da área médica relacionada.

Um sistema de análise de dados socioeconômicos envolve informações de natureza geográfica, que são armazenados sob a forma de dados multidimensionais. Um framework que se propõe a construir sistemas desse tipo e prover métodos para geração de novas informações a partir das existentes requer conhecimentos sobre, além dos temas mencionados, processos de mineração de dados. Dado isto, as sessões seguintes explanam estes conceitos e processos relacionados com a problemática deste trabalho.

2.1. Arquitetura de Framework

Um framework de software é um conjunto de código-fonte ou bibliotecas que tem como objetivo prover funcionalidades comuns a uma classe de aplicações (DOCFORGE, 2013)[1]. Frameworks são usados na criação de softwares que atendem objetivos mais específicos, para isso o usuário deve injetar seu próprio código-fonte a fim de adaptar as funcionalidades genéricas fornecidas pelo framework utilizado.

Por exemplo, um framework de aplicações web pode fornecer ao usuário componentes de gerenciamento de sessão, armazenamento de dados e sistemas de templates de páginas web; de maneira semelhante um framework para aplicações desktop deve possuir códigos prontos de elementos GUI (*Graphical User Interface*) mais comuns.

Ao contrário de bibliotecas ou aplicações tradicionais, o fluxo de funcionamento do programa não é controlado por uma entidade chamadora (ou inicializadora) programada pelo próprio usuário, e sim pelo framework (DIRK, 2000). Mesmo dentro de uma mesma classe de softwares encontramos frameworks que exercem níveis de controle da aplicação bem

[1] <http://docforge.com/wiki/framework>

diferentes: alguns oferecem apenas algumas rotinas básicas ou de utilidades individuais para o shell básico de uma aplicação, outros oferecem quase uma aplicação complexa inteira e exigem rigorosa organização do código-fonte e o cumprimento de regras.

Aplicações desenvolvidas com frameworks possuem uma série de vantagens:

- Usar o código que já foi construído, testado e usado por outros programadores aumenta a confiabilidade e reduz o tempo de programação. Em uma corporação esta reutilização de código efetivamente economiza dinheiro.
- Equipes de desenvolvimento de software podem ser divididas entre aqueles que programam no framework e aqueles que programam a aplicação final. Esta separação de tarefas permite que cada equipe se concentre em objetivos mais específicos e seja melhor aproveitada em suas forças individuais. Por exemplo, os programadores que são especialistas em design de interface podem trabalhar na aplicação do cliente, enquanto os especialistas em segurança, servidor e banco de dados trabalham na estrutura sobre a qual o aplicativo é construído.
- Frameworks podem fornecer recursos de segurança e isso remove o tempo e o custo extra de desenvolvê-la em todas as aplicações que o utilizam.
- Tarefas de baixo nível (como por exemplo conexão com banco, funções de leitura e escrita de arquivo, etc.) geralmente são tratadas pelo framework deixando somente a lógica da aplicação para o programador desenvolver.
- Frameworks muitas vezes ajudam na melhoria das práticas de programação por impor regras ao desenvolvimento do software.

Entretanto, o uso de um framework no desenvolvimento de uma aplicação também acarreta em algumas consequências:

- Muitas vezes a performance do sistema pode ser diminuída. O código genérico do framework pode ser mais lento se comparado ao código otimizado para uma situação específica.
- Frameworks muitas vezes necessitam de um processo de educação significativo para o uso eficiente e correto (ou seja, alguns têm uma alta curva de aprendizado).
- Bugs e problemas de segurança em um framework podem afetar qualquer aplicação que o utiliza. Por isso, deve ser testado e corrigido separadamente ou em conjunto com o produto final de software.

2.2. Sistemas de Informação Geográfica

Sistemas de computador são comumente usados com objetivo de auxiliar na resolução de problemas envolvendo situações do mundo real, para isso os softwares fazem uso de estruturas de dados para armazenar as informações necessárias à resolução dos problemas, como por exemplo, uma matriz numérica para representar uma planilha financeira de gastos. De maneira análoga se o problema é de natureza geográfica os sistemas computacionais devem usar estruturas específicas para o processamento de dados geográficos: para cada objeto geográfico é necessário armazenar seus atributos e as várias representações gráficas associadas devido a ampla gama de aplicações que podem utiliza-la (CÂMARA; DAVIS, 2001).

Quando um sistema é usado para armazenar, analisar e manipular dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente à informação diz-se que este é um Sistema de Informação Geográfica (SIG) (CÂMARA *et al*, 1996 *apud* ARONOFF, 1989; BULL, 1994). É importante ressaltar que sistemas que processam dados espaciais, isto é, dados que possuem uma localização definida, podem não ser considerados SIGs: a característica "geográfica" da informação indica que ela refere-se a um dado da superfície terrestre ou próxima dela, enquanto "espacial" refere-se a qualquer espaço, apesar disso, técnicas de SIG vêm sendo aplicadas em espaços não-geográficos, como superfícies de outros planetas, espaço sideral e até mesmo sobre o espaço do corpo humano em sistemas de análise de imagens médicas (LONGLEY *et al*, 2005).

As aplicações SIG mais tradicionais são as ambientais, tais como sistemas de informação de solos ou de estudo ambiental de mudanças globais; e aplicações socioeconômicas, como sistemas para serviço de utilidade pública, de informações sobre a terra e de censo (CÂMARA *et al*, 1996). Porém o domínio de aplicações SIG está se ampliando cada vez mais acompanhando a evolução dos dispositivos de coleta e as facilidades computacionais em geral, nos últimos anos isto se tornou mais acentuado com a popularização de dispositivos com GPS, tais como, smartphones e navegadores automotivos, permitindo novas classes de aplicações SIG, como por exemplo, aplicações de informações de trânsito, de transporte público, turismo e geomarketing.

Em uma visão geral, pode-se dizer que um SIG possui cinco componentes básicos:

1. Interface com o usuário: que define como o sistema é operado e controlado.
2. Entrada e integração dos dados: que realiza o processamento dos dados

geográficos.

3. Funções de consulta e análise espacial.
4. Visualização e plotagem: exibe os mapas e a representação da informação sobre eles.
5. Armazenamento e recuperação de dados: organizados sob a forma de um banco de dados geográfico.

Cada sistema, em função de seus objetivos e necessidades, implementa esses componentes de forma distinta, mas todos estão usualmente presentes em um SIG (CÂMARA; DAVIS, 2001).

2.3. Modelagem Multidimensional

A tecnologia mais usada pelos softwares quando necessitam armazenar e recuperar um médio ou grande volume de dados é através dos bancos de dados relacionais. Tais bancos são construídos baseados em modelos de dados relacionais, no quais todos os dados estão dispostos em tabelas e sua definição teórica e fundamentada na lógica de predicados e na teoria dos conjuntos. São construídos dessa forma no intuito de prover o acesso às informações de maneira ágil e para possibilitar uma grande variedade de abordagens no tratamento das informações (ROHDEN, 2009).

A modelagem relacional é capaz de representar e oferecer solução para grande parte das situações que envolvem o uso de banco de dados, entretanto para classes de problemas que envolvem quantidades enormes de dados e processamento analítico surgiu o conceito de Data Warehouse. É um pouco difícil formular uma definição rigorosa para Data Warehouse, pois pode ser definido de diversas maneiras (HAN; KAMBER, 2006): pode ser entendido como um conjunto de dados orientado por assunto, integrado, variável com o tempo e não-volátil, que fornece suporte a processos de tomada de decisão (HAN; KAMBER, 2006 *apud* INMON, 1993); pode ser entendido também como um processo de integração de dados em um único repositório e é também um ambiente de suporte à decisão que alavanca dados armazenados em diferentes fontes, organiza-os e entrega-os aos tomadores de decisões (ROHDEN, 2009 *apud* SINGH, 2001).

Data Wirehouses e sistemas OLAP (*On-line Analytical Processing*) são baseados em modelos de dados multidimensionais. O modelo multidimensional é formado por uma tabela central (tabela de fatos) e várias outras a ela interligadas (tabelas de dimensão), sempre por

meio de chaves especiais, que associam o fato às dimensões a qual pertencem (ROHDEN, 2009).

Alguns conceitos importantes dentro da modelagem multidimensional são:

1. Fatos: são os dados a serem agrupados, contendo os valores de cada medida para cada combinação das dimensões existentes. O tamanho da tabela que contém os fatos merece atenção especial do analista.
2. Dimensões: estabelecem a organização dos dados, determinando possíveis consultas/cruzamentos. Por exemplo: região, tempo, canal de venda,... Cada dimensão pode ainda ter seus elementos, chamados membros, organizados em diferentes níveis hierárquicos.
3. Medidas: são os valores a serem analisados, como médias, totais e quantidades.
4. Agregações: totalizações calculadas nos diversos níveis hierárquicos.

Para a representação de dados multidimensionais é comumente usado os esquemas de dados em estrela, floco de neve e constelação.

O esquema em estrela é o paradigma de modelagem mais comum, em que o fato é o centro do dado e ele está ligado diretamente a todas às dimensões que o compõe; sua representação pode ser vista como um gráfico radial (Figura 2.3.1) onde as dimensões encontram-se nas extremidades. As principais vantagens do esquema em estrela é que é bem simples de ser trabalhado e não permite duplicidade, entretanto há algum desperdício de espaço já que é necessário gravar todas as relações que o dado possui com suas dimensões.

O esquema em floco de neve é uma variação do esquema em estrela, no qual as dimensões do dado são normalizadas permitindo, dessa forma, que as dimensões do fato tenham dados relacionados; sua representação pode ser vista como um grafo (Figura 2.3.2). É adequado quando as dimensões do dado têm uma representação composta e economiza espaço ao reduzir a redundância, entretanto é um esquema mais complexo de ser trabalhado. Já o esquema em constelação é quando fatos representados em esquema estrela compartilham dimensões entre si. Pode-se observar na Figura 2.3.3 como as estrelas podem compartilhar dimensões entre si.

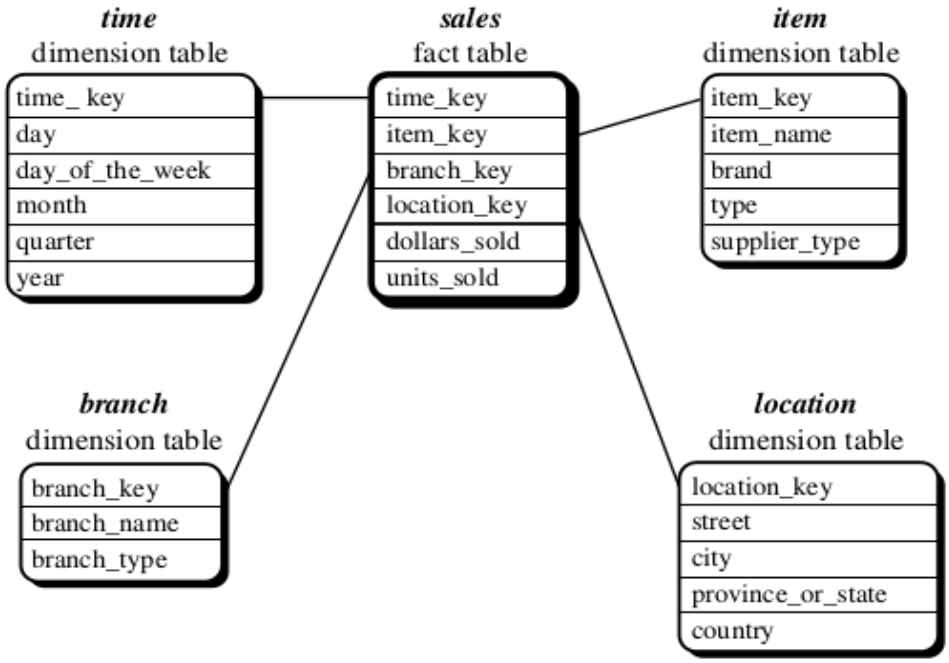

Figura 2.3.1: Exemplo de esquema em estrela de tabelas para um sistema de vendas.

Fonte: Han & Kamber, 2006.

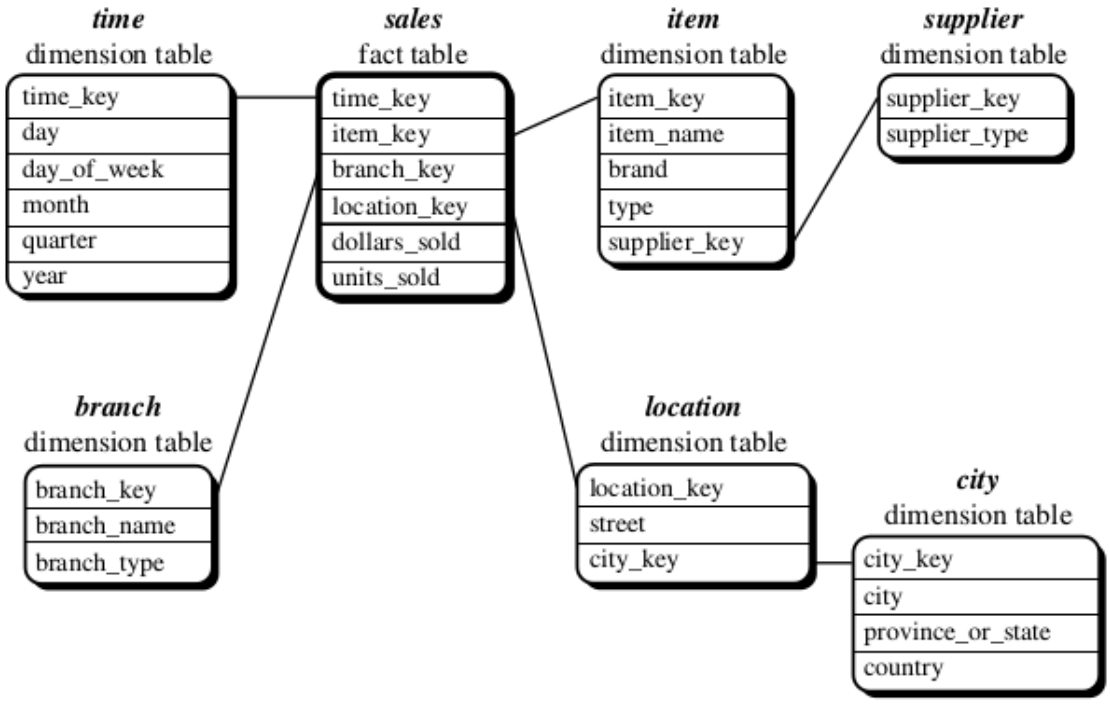

Figura 2.3.2: Exemplo de esquema em floco de neve de tabelas para um sistema de vendas.

Fonte: Han & Kamber, 2006.

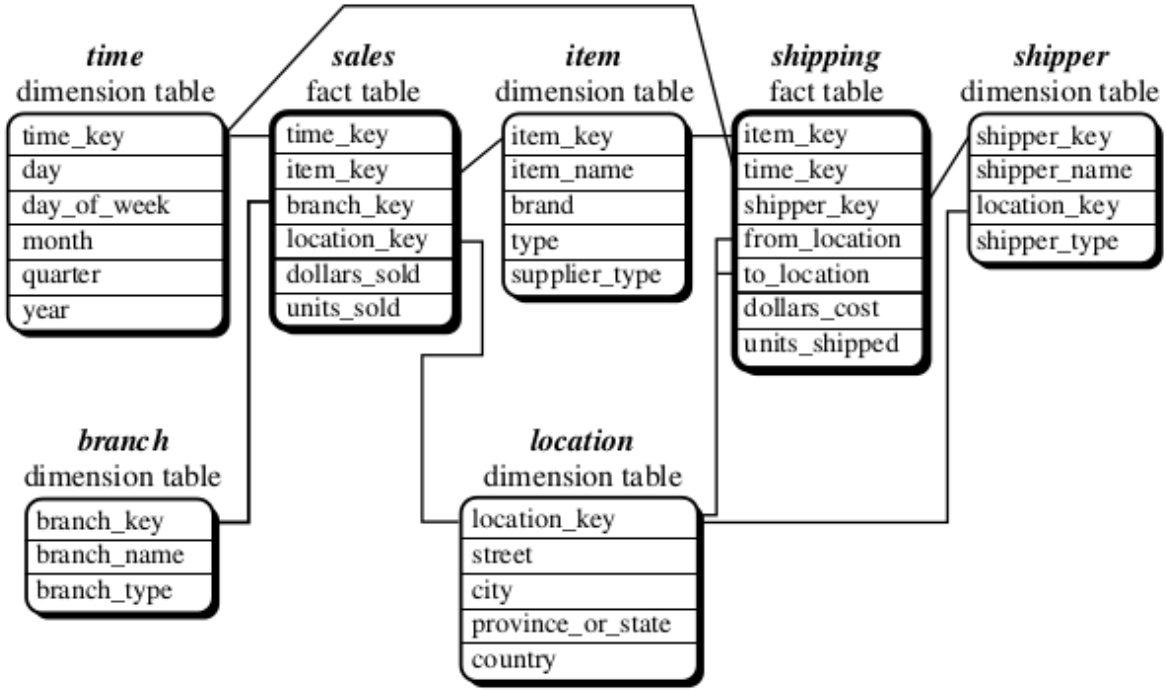

Figura 2.3.3: Exemplo de esquema em constelação de tabelas para um sistema de vendas.
Fonte: Han & Kamber, 2006.

2.4. Mineração de Dados

Ao processo de extrair ou "minerar" conhecimento sobre uma grande quantidade de dados dá-se o nome Mineração de Dados (HAN; KAMBER, 2006). Nos anos recentes a indústria e a sociedade têm sido atraídas pela mineração de dados devido à produção de conhecimento e informação útil através desse processo. O comércio eletrônico, por exemplo, vem pressionando drasticamente as empresas para utilização de formas mais elaboradas de obtenção de conhecimento sobre seus clientes.

Igualmente no setor público a introdução do e-governo visa potencializar o acesso da população às diversas instâncias governamentais, situação que proporciona a geração de dados referentes à preferência política dos eleitores e dos problemas que necessitam atenção e intervenção do governo. Esses dados, inicialmente em forma maciça precisam ser minerados para obtenção de conhecimento e satisfazer os objetivos das empresas ou organizações que se interessam. Presidentes de grandes corporações como IBM, Microsoft e Harley-Davidson não foram capazes de prever que o mercado ia preferir PC's, Internet e motos populares (BRAGA, 2005).

A mineração de dados está inserida em um processo maior chamado Descoberta de Conhecimento em Banco de Dados (*Knowledge Discovery in Database – KDD*). A saber, a mineração de dados se restringe à obtenção dos modelos; as etapas anteriores a esse processo e a própria mineração de dados são instâncias da KDD que, geralmente, é usada num banco de dados multidimensional devido ao grande volume de dados armazenados.

Um projeto de mineração de dados segue uma sequência de etapas para obtenção do conhecimento. Tais etapas são:

- Definição do problema: Envolve descobrir realmente qual a informação que se deseja obter. A resposta a uma pergunta mal formulada fica comprometida desde o início do processo. Esta etapa especifica a porção do banco de dados que o usuário está interessado.
- Aquisição e avaliação dos dados: assumindo que a informação está nos dados selecionados na etapa anterior, eles são coletados e formatados. Aplicar conceitos de hierarquia nos dados é uma forma comum de formatação, pois permite que os dados sejam minerados em múltiplos níveis de abstração. A formatação também pode especificar que relação os dados têm uns com os outros.
- Extração de características e realce: após a formatação do dado se identifica quais atributos dele contribuem para a resolução do problema. Nessa etapa é produzido um conjunto de dados (data set) representativo, reproduzível e confiável.
- Desenvolvimento do modelo e implementação: desenvolvem-se os modelos descritivos ou preditivos, dependendo do objetivo da mineração, e entrega-se o produto final representando a informação descoberta através de padrões que podem ser tabelas, gráficos, mapas, árvores de decisão, cubos, etc.

Diversas ferramentas foram desenvolvidas no intuito de tornar a aplicação da Mineração de Dados uma tarefa menos técnica, e com isto possibilitar que profissionais de outras áreas possam fazer usa dela , um bem conhecido é o Weka (*Waikato Environment for Knowledge Analysis*), desenvolvido na Universidade de Waikato, na Nova Zelândia (WEKA, 2013)[2]. Ele possui implementações de algoritmos de aprendizagem máquina, que podem ser facilmente aplicados a qualquer data set, e métodos para os problemas de mineração de dados mais tradicionais, permitindo gerar hipóteses para a solução de problemas a partir dos padrões encontrados nos dados (WITTEN; FRANK, 2005). Weka é uma das melhores ferramentas

[2] <http://cs.waikato.ac.nz/ml/weka>

livre (CAMILO; SILVA, 2009) foi inicialmente escrito em Java e é distribuído sob a GNU General Public License. Os algoritmos podem ser aplicados diretamente na ferramenta através da interface gráfica, a qual os desenvolvedores deram o nome *Explorer* (Figura 2.4.1).

Figura 2.4.1: Interface Explorer do pacote de software Weka.

Fonte: Witten & Frank, 2005.

Entre os softwares proprietários destacam-se Clementine (NISBET; ELDER; MINER, 2009) – uma das ferramentas líder de mercado, desenvolvida pela SPSS, posteriormente incorporada à IBM (IBM, 2013)[3]; a ferramenta SAS Enterprise Miner Suite (SAS, 2013)[4] – desenvolvida pela empresa SAS, é uma dos mais conhecidos softwares para mineração de dados; e Oracle Data Mining (ORACLE, 2013)[5] - ferramenta para a mineração de dados desenvolvida para o uso em bancos de dados Oracle (CAMILO; SILVA, 2009).

[3] <http://ibm.com/software/analytics/spss>
[4] <http://sas.com/technologies/analytics/datamining>
[5] <http://oracle.com/technetwork/database/options/advanced-analytics/odm>

3. SISTEMAS DE ANÁLISE DE DADOS SOCIOECONÔMICOS

No cenário nacional existem algumas aplicações que, assim como o framework que será proposto nesse trabalho , também buscam oferecer meios para que o público-alvo tenha acesso à informação sobre indicadores socioeconômicos e possa analisá-los de acordo com objetivos específicos. Na maioria dos casos a informação encontra-se georreferenciada permitindo a análise por meio de mapas.

3.1. Sistema IBGE de Recuperação Automática – SIDRA

O SIDRA é o sistema do Instituto Brasileiro de Geografia e Estatística (IBGE) para consulta dos indicadores e variáveis socioeconômicas. Com o SIDRA é possível consultar dados na forma de séries temporais, acompanhando seu comportamento ao longo do tempo, bem como ter os mesmos disponibilizados por níveis territoriais desagregados, como município, distrito e bairro, de modo a facilitar o conhecimento da realidade municipal (IBGE, 2013)[6].

Esse sistema funciona sob a plataforma web, podendo ser acessado através de navegadores de internet através do endereço: sidra.ibge.gov.br. É capaz de gerar visualizações em modo de tabelas (Figura 3.1.1), gráficos (Figura 3.1.2) e em mapas (Figura 3.1.3) contendo uma base de dados com mais de seiscentos milhões de variáveis.

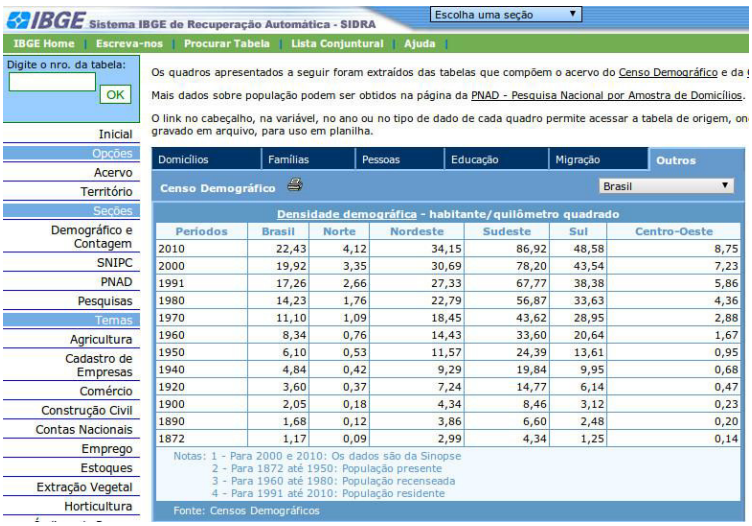

Figura 3.1.1: SIDRA exibindo a variável "Densidade Demográfica" em forma tabular. Fonte: IBGE, 2013

[6] <http://www.ibge.gov.br/home/disseminacao/eventos/workshop/sidra.shtm>

Atualmente, as variáveis estão organizadas em dezenove temas, tais como Agricultura, Cadastro de Empresas, Construção Civil, População, entre outros.

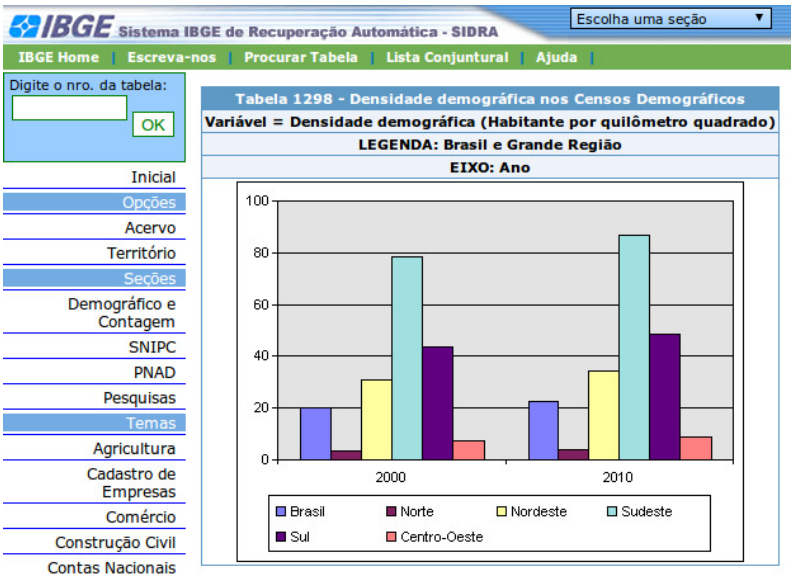

Figura 3.1.2: SIDRA exibindo a variável "Densidade Demográfica" em forma gráfica.

Fonte: IBGE, 2013

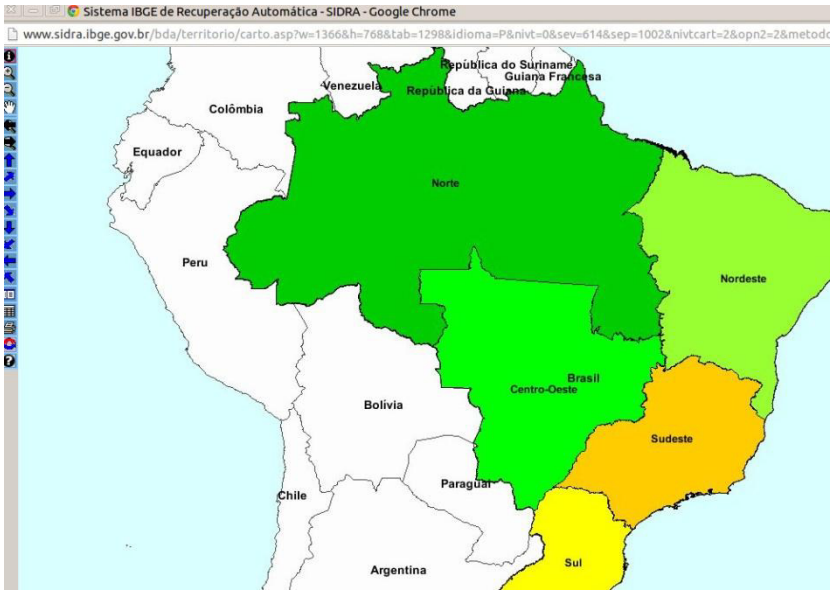

Figura 3.1.3: SIDRA exibindo a variável "Densidade Demográfica" em forma de cartograma.

Fonte: IBGE, 2013

A geração dos relatórios em tabela permite que sejam definidos filtros de local e ano para os dados que serão exibidos e, a partir do relatório tabular, pode ser gerado o relatório gráfico ou em mapa.

É uma das propostas do SIDRA a fácil reprodução do conteúdo tabular por qualquer órgão de administração pública para disseminação de seus dados agregados, além disso é

possível o envio de dados por e-mail a partir de agendamentos.

3.2. DevInfo do Portal ODM

O DevInfo v.6.0 é um software desenvolvido com a cooperação do sistema das Nações Unidas e é adaptado da tecnologia Childinfo da UNICEF (CHILDINFO, 2013)[7]. O Portal ODM, onde o software DevInfo se encontra, é uma ferramenta web para acompanhar a situação dos Objetivos de Desenvolvimento do Milênio (ODM). Os ODM são metas pactuadas pelo Brasil e por outros 190 países membros das Nações Unidas para melhorar indicadores sociais, ambientais e econômicos.

O objetivo do Portal ODM é permitir que o cidadão possa acompanhar a realidade de seu município e envolvê-lo no processo de implementação de políticas públicas. Ao mesmo tempo, as empresas poderão contar com informações para a definição de suas ações de responsabilidade social (Portal ODM, 2013)[8].

Um diferencial no DevInfo é a possibilidade de definição de filtros para os relatórios sobre três dimensões: ano, local e fonte dos dados, sendo assim os relatórios gerados são totalmente dinâmicos. É capaz ainda de exibir a informação em forma tabular, em gráfico e em mapa (Figura 3.2.1).

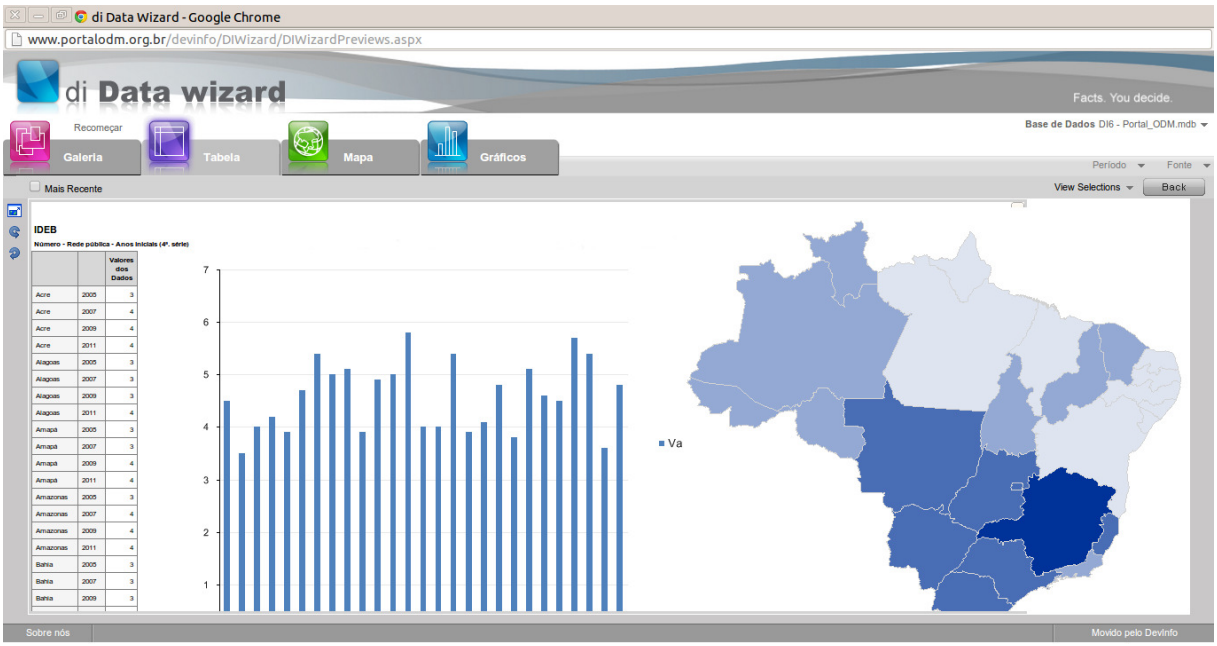

Figura 3.2.1: Compilação dos três modos de visualizações no DevInfo do Portal ODM

Fonte: Portal ODM, 2013

[7] <http://www.childinfo.org>

[8] <http://www.portalodm.com.br/sobre>

3.3. Sistema de Indicadores do Observatório Social do Movimento Nossa São Luís

Esse sistema reúne um conjunto de 75 indicadores sobre a situação da cidade de São Luís do Maranhão e o desempenho das políticas públicas. Os dados são apresentados com séries históricas e, sempre que possível, comparando os resultados de São Luís com as outras capitais brasileiras. Desse total, 15 indicadores são apresentados de forma intraurbana, isto é, por região da cidade, evidenciando as desigualdades internas (Nossa São Luís, 2013)[9].

É um portal web acessível ao público através do endereço eletrônico nossasaoluis.org.br que gera relatórios de indicadores socioeconômicos, alguns em forma de tabelas, outros em forma de mapas (Figura 3.3.1) não dando muita flexibilidade ao usuário para escolher o modo de visualização: o portal possui alguns modelos de relatório pré-definidos e o usuário pode escolher entre eles. Os filtros que podem ser aplicados também são pré-definidos para cada relatório.

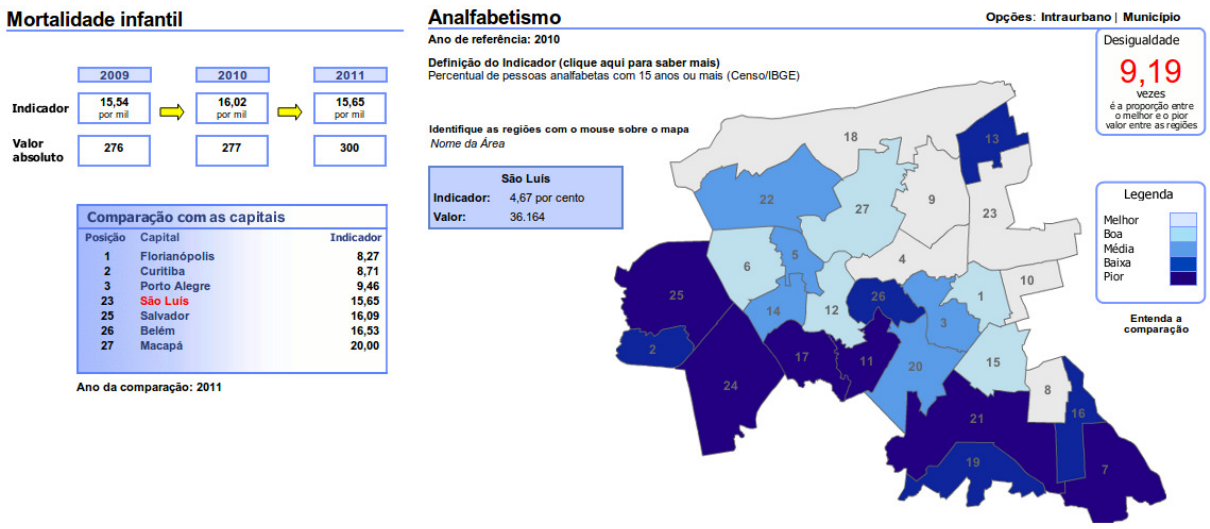

Figura 3.3.1: Compilação de dois relatórios do Sistema de Indicadores do Observatório Social
Fonte: Nossa São Luís, 2013

3.4. Outros portais governamentais para consulta de indicadores socioeconômicos

Muitos órgãos governamentais, na falta de um sistema adequado para disponibilizar relatórios socioeconômicos à população em geral, optaram por apenas publicá-los de forma estática em seus respectivos portais web, seja através de documentos de texto e gráficos prontos, como é o caso do portal da Secretaria de Estado e Segurança Pública do Maranhão (www.ssp.ma.gov.br), ou através da seleção do relatório submetendo a escolha por meio de

[9] <http://www.nossasaoluis.org.br/indicadores2012/apresentacao.html>

formulário web, como por exemplo, a página de estatísticas do Departamento Estadual de Trânsito do Maranhão - DETRAN (www.detran.ma.gov.br).

3.5. Visão geral dos sistemas de análise socioeconômica

Comparando as características presentes em cada um dos sistemas apresentados pode-se perceber, em alguns pontos, deficiências de funcionalidades que poderiam ser muito úteis ao usuário, tanto na questão na obtenção da informação quanto na representação da mesma. Em outros aspectos, foram implementados componentes bem elaborados, que merecem destaque por serem bem aplicados em seus contextos. Em todo caso, a falta de um padrão de software dessa natureza faz com que cada organização desenvolva seu próprio sistema de consulta e representação da informação socioeconômica o que, na maioria dos casos apresentados, gerou um subaproveitamento da base de dados e/ou carência na exibição da informação.

O framework que está sendo proposto nesse trabalho será capaz de gerar sistemas que agregam os principais pontos positivos de cada um dos sistemas mencionados e prover novas funcionalidades:

4. PROPOSTA DE FRAMEWORK

Dados georreferenciados de natureza socioeconômica são computacionalmente complexos, pois as estruturas de dados necessárias para representá-los de forma consistente precisam levar em consideração diversos fatores, tais como a unicidade, a atomicidade e a velocidade de acesso aos dados, além da possibilidade de fragmentação e cruzamento dos mesmos, tomando cuidado também com o espaço gasto para a armazenamento.

Essas estruturas de dados também precisam ser projetadas de forma a facilitar a integração de um software capaz de operá-los para realizar as operações básicas de inserção, edição, recuperação e exclusão dos dados sem comprometer a integridade da base de dados como um todo. Com relação a essas operações básicas, a recuperação dos dados é a que merece maior destaque já que deve prever os casos de exibição dos dados de forma simples (atômica), agrupada (de relatório) e cruzada (gerada).

No intuito de prover tais características e funcionalidades comuns aos sistemas de processamento de dados socioeconômicos georreferenciados este trabalho propõe um framework que realize o tratamento dessas problemáticas de forma genérica e ágil, e que sirva para o desenvolvimento de, desde simples sistemas de geração de relatórios socioeconômicos, a até sistemas complexos de processamento e mineração de dados desse tipo.

Com este objetivo o framework é criado utilizando os conceitos de modelagem multidimensional, para a representação e armazenagem dos dados socioeconômicos; modelagem de mapeamento objeto-relacional, para a interligação do banco de dados com o sistema; mineração de dados, para processamento e construção de dados socioeconômicos; e arquitetura de software divida em modelos, visões e controladores, para prover liberdade ao desenvolvedor para construir e adaptar seu software através do framework.

4.1. Modelagem dos Dados

Esta seção apresenta as decisões tomadas sobre o esquema adotado de banco de dados assim como sua organização.

4.1.1. Esquema de Dados em Estrela

As dimensões de um dado são as características que o tornam diferente de todos os

demais, ou seja, garantem a propriedade de unicidade do mesmo. Um exemplo de dados unidimensionais seria a lista das idades dos alunos de uma turma, cada item dessa lista é único, pois se refere à idade de um aluno diferente; neste caso o aluno é a dimensão do dado. De forma semelhante, um exemplo de dados bidimensionais seria a tabela com as notas de provas dos alunos dessa turma, pois cada item da tabela se refere a uma nota de um aluno e/ou prova diferentes; nesse caso o aluno e a prova são as dimensões do dado.

Os dados socioeconômicos georreferenciados são organizados em pelo menos três dimensões principais: dimensão temporal, dimensão espacial ou geográfica e a fonte do dado. Outras dimensões também podem ser relacionadas dependendo da aplicação, como por exemplo, categoria e unidade de medida. Esse tipo de dados, portanto, é multidimensional. Um dado multidimensional é organizado ao redor do tema central ao qual chamamos de fato. É o valor do fato representado que está ligado às dimensões do dado. No caso dos dados socioeconômicos o fato é chamado de indicador socioeconômico. Um exemplo de indicador socioeconômico seria o número de eleitores de uma cidade; exemplo de valores para suas três dimensões principais poderiam ser "São Luís (MA)" para a dimensão espacial, "ano de 2009" para dimensão temporal e "IBGE" para a fonte do dado.

O paradigma de modelagem de dado em estrela é o mais adequado para a representação de um indicador socioeconômico, pois as propriedades de unicidade e simplicidade de acesso facilitam o gerenciamento dos dados e agilizam a recuperação dos mesmos pelo framework proposto. Além disso, como é esperado que múltiplos indicadores sejam cadastrados, isso gerará um modelo em constelação através do compartilhamento das dimensões dos indicadores, tornando possível o cruzamento de informações para a mineração desses dados. O cenário descrito é exemplificado na Figura 4.1.1.

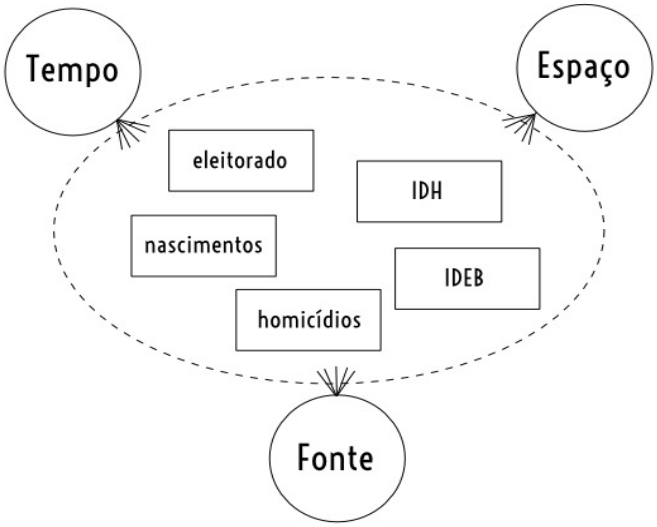

Figura 4.1.1. Indicadores distintos compartilhando os mesmos valores de dimensões.

4.1.2. **Modelagem dos Dados Multidimensionais**

Um modelo multidimensional relaciona as tabelas de fatos com as tabelas de dimensões, possibilitando inúmeras combinações. A modelagem proposta por este framework é realizada de forma que permita a mineração em diversos níveis de complexidade dos indicadores socioeconômicos sem que haja perda de performance nas consultas.

As dimensões do dado devem ser armazenadas da seguinte maneira: se a dimensão for monovalorada então ela deve ser gravada diretamente em uma coluna na tabela do indicador, caso seja multivalorada a dimensão deve possuir uma tabela própria e na tabela do indicador deve ser gravada a chave estrangeira que aponta para o valor correspondente na tabela da dimensão. As colunas de dimensões, sejam do tipo monovalorada ou chave estrangeira, formam a chave primária composta da tabela do indicador socioeconômico, portanto as três dimensões principais, que são o ano, o local e a fonte são a chave primária composta.

O framework proposto não limita o indicador a possuir somente as três dimensões principais supracitadas, caso seja necessário a representação de mais dimensões a mesma deve ser adicionada na chave primária composta segundo a regra da mesma forma que as outras.

Com objetivo de aumentar o desempenho das consultas e reduzir o volume de dados gravados, os indicadores de uma mesma categoria devem ser agrupados na mesma tabela. Se cada indicador pode ser considerado um dado num esquema de estrela então as tabelas de indicadores podem ser vistas como dados num esquema de constelação, pois compartilham das mesmas dimensões (chaves primárias).

Por exemplo, os seguintes indicadores socioeconômicos: número de eleitores analfabetos, número de eleitores que concluíram o ensino médio e número de eleitores com ensino superior devem ser agrupados em uma mesma tabela, pois compartilham da mesma categoria, a saber, número de eleitores, como podemos ver na figura 4.1.2. Já um indicador que representa o número de empregos gerados não deve ser agrupado na mesma tabela, pois, mesmo que coincidam as mesmas dimensões, não estão na mesma categoria.

Obviamente, agrupar fatos numa tabela comum permitirá muitos campos nulos, porém essa desvantagem de desperdício de espaço é necessária para que se tenha um ganho de performance. O objetivo principal desse modelo é permitir a geração instantânea de relatórios e isso é possível deixando o dado o menos fragmentado possível.

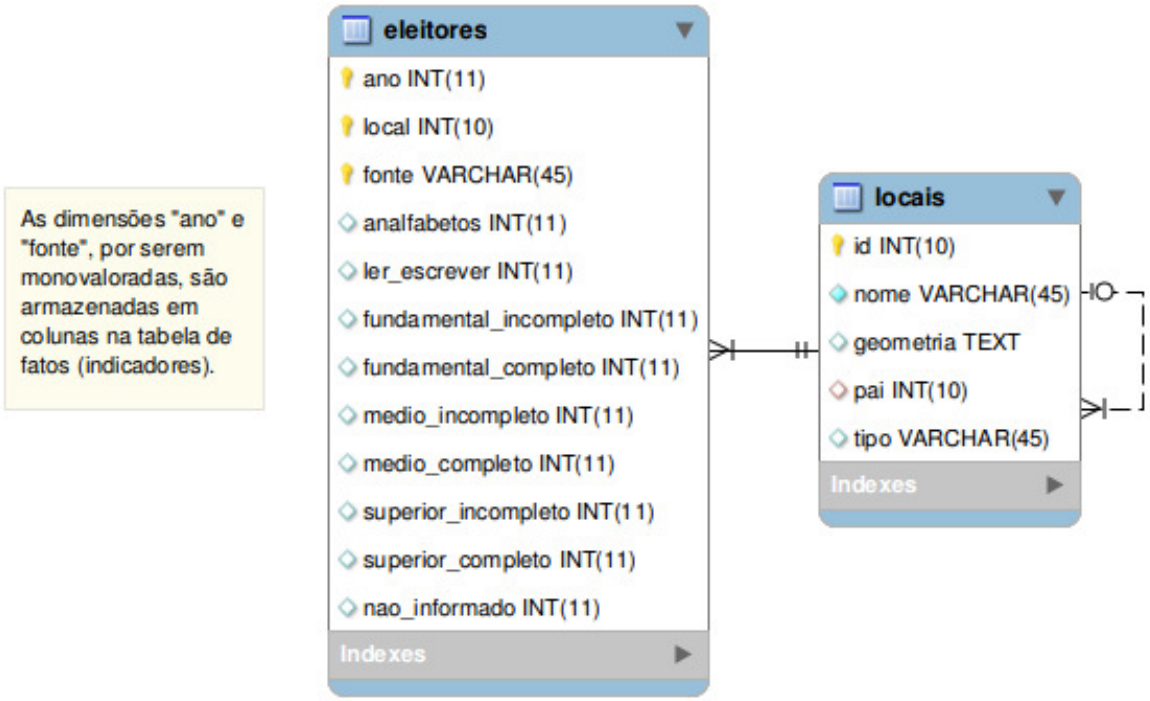

Figura 4.1.2: Tabelas referentes aos indicadores de eleitorado.

4.1.3. **Tabelas de Lookup do Framework**

Para que o framework tenha acesso rápido a todos os indicadores socioeconômicos disponíveis no sistema é necessária uma tabela de lookup para armazenagem dessas variáveis. Cada entrada gravada na tabela de variáveis corresponde potencialmente a um relatório que o sistema é capaz de gerar e deve ter a seguinte estrutura:

- Possuir uma chave primária composta de duas colunas, a primeira é o nome da tabela de indicadores, e a segunda é a coluna dessa tabela, ou seja, o indicador propriamente dito. Dessa forma cada indicador possui exatamente uma entrada na tabela de lookup que o representa ao mesmo tempo em que o framework tem acesso a informação de onde esse dado está localizado.
- Possuir colunas de metadados do indicador socioeconômico representado. As principais são: o título e a unidade de medida em que o dado está gravado. Também é possível gravar observações a repeito daquele dado ou qualquer outra informação que se julgar necessária.

Como o framework prevê casos de mineração de dados socioeconômicos, é necessária outra tabela de lookup para armazenagem de variáveis geradas através do processo de

mineração de dados. Cada entrada na tabela de variáveis mineradas corresponde a um relatório que o sistema é capaz de minerar. Deve ter a seguinte estrutura:

- Possuir uma chave primária simples, como por exemplo, um número de id autoincremental.
- Possuir uma coluna que defina o conjunto de variáveis utilizadas no processo de mineração. Esse conjunto de variáveis pode referenciar qualquer variável previamente cadastrada, seja ela um indicador ou outra variável minerada. A possibilidade de minerar dados através de outros dados minerados faz com que as possibilidades de geração de novas informações através desse processo sejam maximizadas e sejam, conceitualmente, sem limites.
- Possuir uma coluna que defina o cálculo a ser empregado no conjunto de variáveis definidas na coluna anterior para obtenção do novo dado.
- Tanto a coluna de variáveis, quanto a coluna do cálculo devem armazenar uma string possível de ser interpretada pela aplicação, como por exemplo, XML, JSON ou uma linguagem de representação própria.
- Possuir as colunas de metadados do indicador socioeconômico minerado tais como título e unidade de medida.

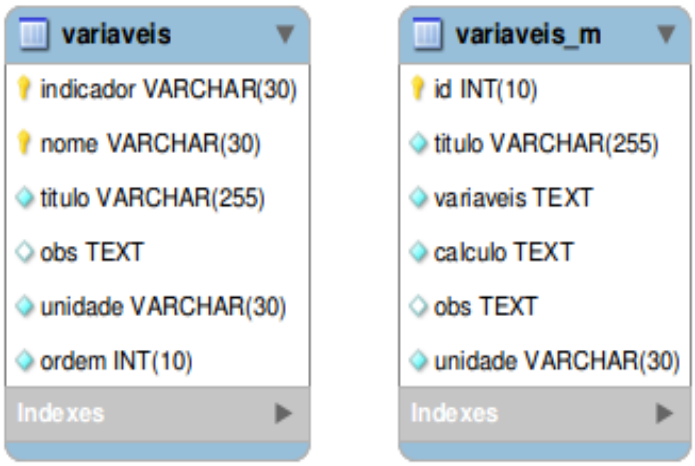

Figura 4.1.3: Tabelas de lookup do framework.

4.2. Arquitetura

Com os esquemas de dados já definidos foi preciso estabelecer uma arquitetura para o framework que fosse flexível em relação às diversas possibilidades de exibição das

informações. Softwares de análises de informações socioeconômicas geralmente oferecem três tipos de visualizações dos dados: de forma tabular, em gráficos e em mapas, portanto o framework não deveria está ligado a nenhuma forma de exibição específica, em vez disso, deveria prover uma interface genérica para qualquer método de visualização.

A arquitetura de software MVC (MINETTO, 2007) revelou-se perfeita para a resolução do problema. Nela há uma forte separação entre a representação da informação, a interação do usuário e o fluxo da aplicação, através da divisão do sistema em três grandes componentes: modelos, que representam os dados da aplicação; as visões, que exibem as informações; e os controladores, que realizam o processamento das entradas e dos dados fazendo a ligação entre os modelos e as visões. Sendo assim o framework se ocupa principalmente com os modelos e os controladores deixando o desenvolvedor livre para criar quaisquer visões que desejar para exibição dos dados.

As subseções seguintes apresentam como foi organizado cada camada para atingir os objetivos do framework proposto.

4.2.1. **Camada de Modelos**

O framework utiliza dois pares de modelos para representação dos dados. O primeiro par refere-se aos indicadores cadastrados no banco: o modelo de mapeamento objeto-relacional (*ORM*) **Variável**, da tabela de lookup de indicadores e o modelo **Relatório**, que é responsável por extrair e montar os dados em forma de relatório através uma instância do modelo Variável. O segundo par de modelos refere-se aos indicadores que podem ser obtidos através de mineração de dados: o *ORM* **Variável_M**, da tabela de lookup de indicadores minerados e o modelo **Relatório_M**, que é responsável por extrair, calcular e montar os dados em forma de relatório através uma instância do modelo Variável_M. O relacionamento estabelecido entre os modelos de relatório e os *ORM*s do framework está representada na Figura 4.2.1.

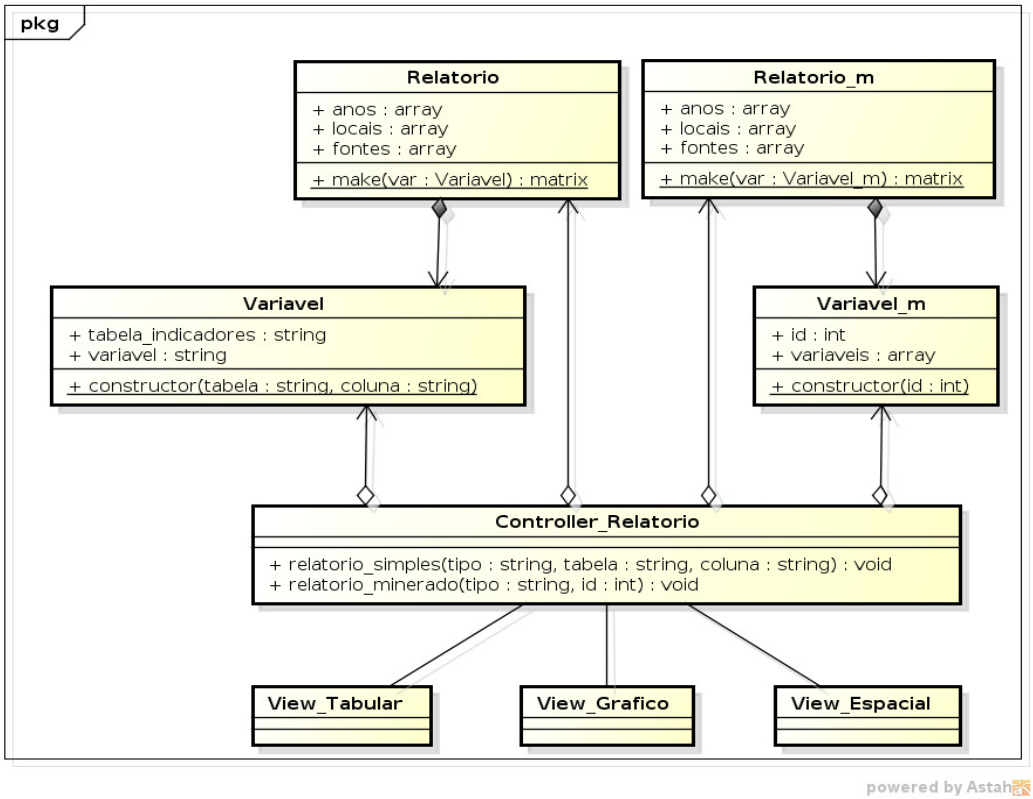

Figura 4.2.1: Diagrama de Classes que mostra os relacionamentos entre as classes de arquitetura MVC do framework.

4.2.1.1. Modelos *ORM* (*Object-relatinal Mapping*)

Quando uma tabela de um banco de dados é representada através de uma classe e os registros dessa tabela, através de instancias dessa classe podemos dizer que esta é uma classe de mapeamento objeto-relacional (*ORM*). Classes *ORM* contam ainda com métodos que realizam operações na tabela representada, dessa forma, comandos SQL como, por exemplo, INSERT, DELETE ou UPDATE são realizados através de uma interface de programação simples que realiza todo o trabalho de persistência, dessa forma o programador não precisa de preocupar com comandos em linguagem SQL.

Os modelos *ORM* Variável e Variável_M permitem a leitura e gravação nas tabelas de lookup de forma rápida, segura e independente da arquitetura de banco de dados, tornando o framework flexível quanto ao Sistema Gerenciador de Banco de Dados (SGBD) a ser utilizado.

4.2.1.2. Modelos de Relatórios

Os modelos de relatórios utilizam os modelos *ORM* para construção da informação na forma de relatório. Eles têm um método principal chamado **make**, que constrói uma matriz

multidimensional com os dados representados na instância do modelo *ORM* passado como parâmetro.

O método make do modelo Relatório realiza consulta (*query*) no banco de dados para extrair todos os dados referentes ao indicador representado pela instância de Variável passada como parâmetro. É possível ainda informar parâmetros de filtro para que nem todos os dados sejam retornados, somente aqueles cujos valores das dimensões sejam permitidos pelos filtros definidos. Essa sequência de passos pode ser observada na Figura 4.2.1.1.

Após a extração de dados do banco o método make se encarrega de gerar a matriz multidimensional com os dados adquiridos. O número de dimensões dessa matriz é igual ao número de dimensões que o indicador possui, assim, se as dimensões forem ano, local e fonte, será gerada uma matriz tridimensional. Cada elemento dessa matriz será o dado indexado pelos três valores das dimensões a qual pertence.

O método make do modelo Relatorio_M se difere do anterior apenas no que diz respeito à complexidade. Este realiza a consulta no banco, não somente de um indicador, mas de tantos quanto forem definidos no conjunto necessário para o processo de mineração na instância da Variavel_M passada como parâmetro. Opcionalmente esse método também pode aceitar filtros.

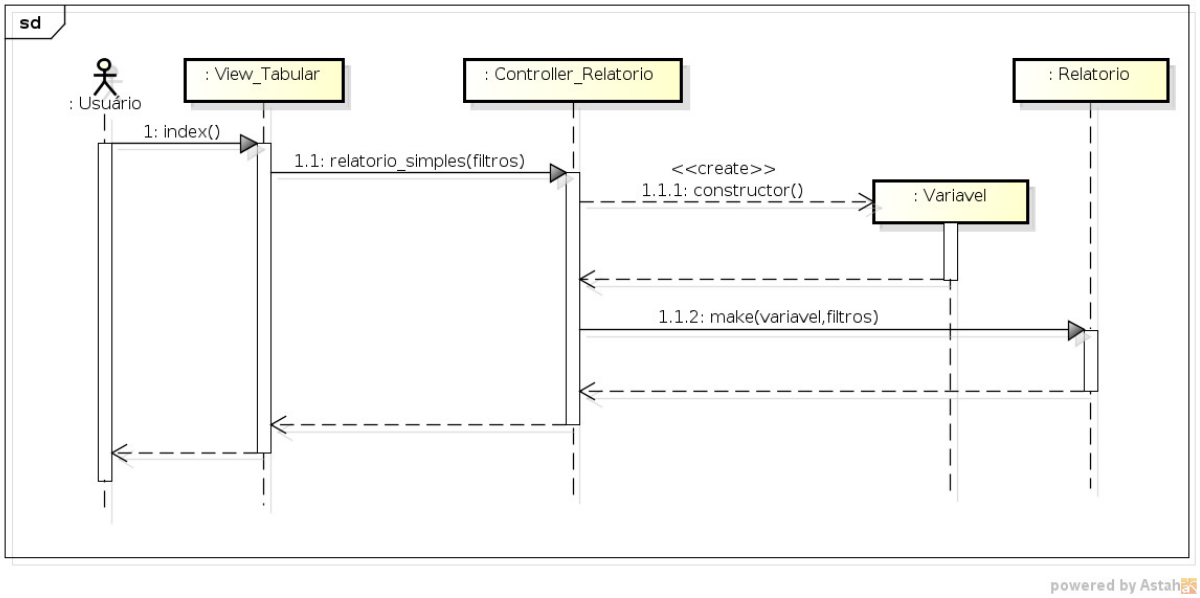

Figura 4.2.1.1: Diagrama de sequência do caso de geração de relatório tabular.

Após a extração dos dados de todos os indicadores o método make aplica a função de mineração definida. A função de mineração é o cálculo para cada indicador extraído que gerará o relatório propriamente dito. Só após o cálculo ser realizado sobre todos os dados é

que a matriz multidimensional será gerada (Figura 4.2.1.2).

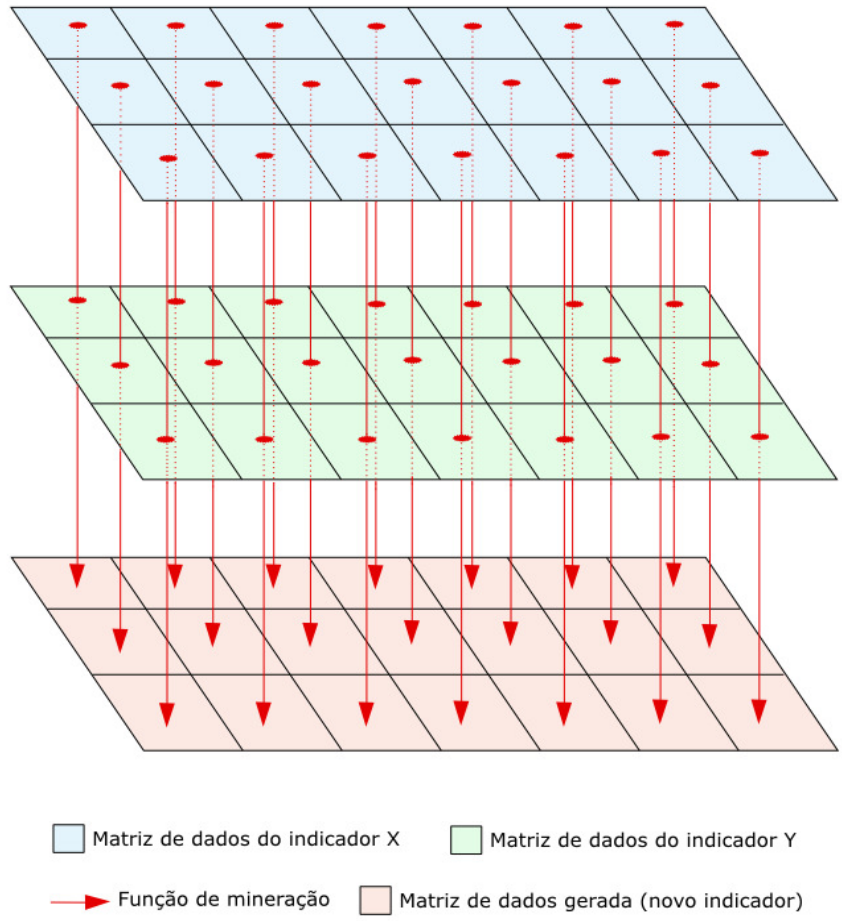

Figura 4.2.1.2: Método make executando a função de mineração.

A função de mineração ou cálculo pode utilizar as operações aritméticas básicas (adição, subtração, multiplicação e divisão), além das funções matemáticas mais comuns como exponenciação, logaritmo, módulo, resto, máximo, mínimo, etc., usar constantes nos cálculos e representar funções matemáticas definidas por partes, isto é, que depende de uma condição, desde que siga a estrutura SE-SENÃO.

A função de mineração também deve proporcionar meios para que os cálculos possam ser realizados entre células da matriz de dimensões diferentes, isto é, em um grau deslocado de relacionamento (Figura 4.2.1.3). Em uma situação normal a função de mineração seria aplicada entre valores de indicadores diferentes, mas somente se eles tivessem os mesmos valores de dimensões. Por exemplo, somando o “número de eleitores analfabetos de São Luís em 2013” com “número de eleitores escolarizados de São Luís em 2013” resulta em “total de eleitores de São Luís em 2013”, porém se quiséssemos comparar o crescimento do número de eleitores analfabetos entre 2012 e 2013 precisaríamos calcular a subtração entre células de

valores de dimensões temporais diferentes.

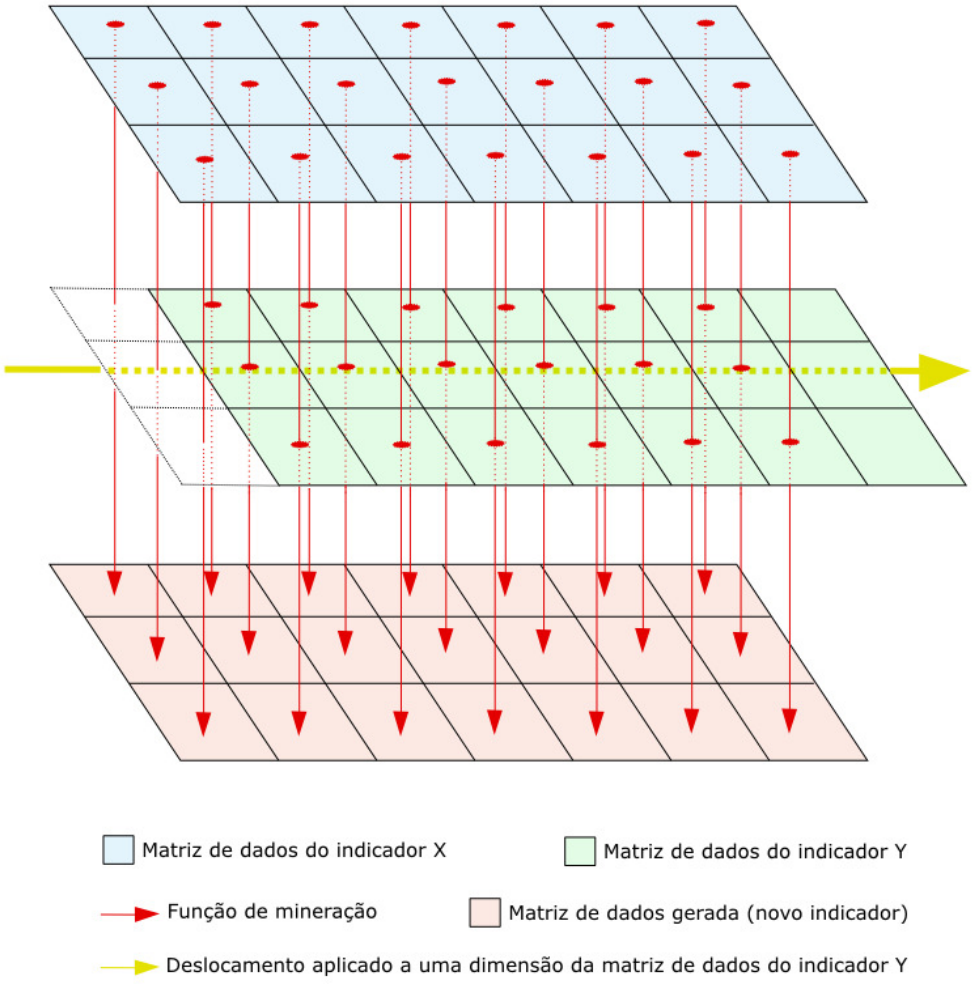

Figura 4.2.1.3: Método make executando a função de mineração com deslocamento dimensional.

4.2.2. **Controladores**

Os controladores do framework recebem as informações vindas das visões e disparam o mecanismo de construção do relatório requerido pelo usuário (Figura 4.2.2). Após receber o pedido de um relatório e, opcionalmente, filtros para o mesmo, o controlador instancia o modelo *ORM* correspondente e realiza a chamada do método make do relatório requerido passando o modelo *ORM* criado juntamente com os filtros (se houver).

Após o término da execução do método make a matriz multidimensional gerada é enviada a uma das visões do sistema onde essa informação será exibida apropriadamente.

Fluxograma 4.2.2: Funcionamento MVC do framework

4.2.3. **Visões**

Não é propósito desse framework definir padrões de visões para os dados socioeconômicos, mas prover uma interface amigável para que o desenvolvedor possa construir suas próprias visões.

Os dados vindos do controlador vêm em forma genérica permitindo de forma simples que ele seja moldado e exibido por uma visão. As formas mais comuns para exibir um dado socioeconômico são em forma de tabelas, de gráficos ou desenhados em mapas.

4.3. Funcionalidades fornecidas pelo framework

Visto estas especificações, um sistema construído através do framework proposto poderá dispor as características de:

4.3.1. **Geração de Relatórios Dinâmicos**

A possibilidade de definição de filtros para a informação a ser exibida e a montagem dos relatórios com os dados atuais armazenados no banco fazem com que os sistemas DevInfo e SIDRA, mencionados no Capítulo 3, gerem relatórios dinâmicos. Quando o relatório é estático, a exemplo do sistema do Movimento Nossa São Luís, o usuário pode receber uma representação da informação com dados desnecessários para ele, dados desatualizados, ou com dados insuficientes, mesmo quando a informação está presente na base de dados.

Sabendo que é de fundamental importância que o usuário possa personalizar a

informação que será representada no sistema, o framework proposto provê meios de filtragem por dimensões do dado (isto é, pode filtrar por tempo, local, fonte, etc.), filtragem pelo valor do dado (algo que não foi constatado nos sistemas mencionados) e combinação de múltiplos filtros. Além disso, o framework sempre retorna ao usuário um relatório atualizado em tempo real com os dados do banco: não usa de cache de relatórios já que foi projetado para retornar a informação de forma ágil mesmo sob um grande volume de dados.

4.3.2. **Múltiplas Representações da Informação**

Todos os sistemas citados anteriormente no Capítulo 3 traziam ao menos duas formas diferentes de visualizar a informação, a saber, tabelas, gráficos ou mapas. O framework proposto é projetado para sempre entregar a informação ao desenvolvedor de maneira genérica, deixando-o livre para aproveitá-la de acordo com suas necessidades. É natural e perfeitamente possível a construção de tabelas, gráficos de qualquer tipo e de mapas com a informação repassada pelo framework. Além disso, como a informação vem de forma não-formatada para um tipo específico de representação, pode-se usá-la para outras necessidades fora desse contexto, como será explicado nas seções seguintes.

4.3.3. **Exportação de Relatórios em Documentos Digitais**

Assim como a representação em forma de tabelas, gráficos ou mapas, o desenvolvedor pode optar por usar os dados retornados pelo framework para montagem de um documento digital utilizando bibliotecas de exportação.

4.3.4. **Integração com Outros Sistemas**

Um sistema construído com o framework proposto pode ser usado como provedor de um WebService de consulta ou inserção na base de dados para outros sistemas. Esta é somente mais uma forma de representar a informação não havendo necessidade de quaisquer alterações no código do framework, disponibilizando facilidade para integração de bases ou ferramentas de extração de informação automatizadas.

4.3.5. **Criação de Novos Indicadores a Partir dos Existentes**

Novos indicadores socioeconômicos podem ser processados e exibidos através da mineração de dados dos indicadores existentes. Essa é uma característica que não está presente nos sistemas governamentais de análise de indicadores, anteriormente mencionados.

É importante ressaltar que o framework não funciona gravando no banco os dados dos indicadores obtidos através do processo de mineração, ele os processa toda vez que são requisitados garantindo que a informação minerada esteja sempre atualizada. O processo de mineração segue uma estratégia de performance otimizada para que a geração de relatórios minerados se dê de forma ágil, tal como os relatórios dos demais indicadores, que estão armazenados em banco de dados.

4.3.6. **Cruzamento de Dados de Indicadores**

O processo de mineração pode ser usado para realizar cruzamento entre dois ou mais indicadores permitindo a comparação dos mesmos. Um indicador minerado é geralmente construído usando cálculos aritméticos sobre os valores de outros indicadores, gerando um valor novo que o representa, entretanto, a função de mineração pode ser usada simplesmente para a exibição lado a lado dos indicadores escolhidos, gerando assim um relatório de caráter comparativo.

4.3.7. **Flexibilidade para Agregação de Dimensões ao Dado**

O framework proposto é projetado para suportar dados armazenados sob um número indefinido de dimensões. Essa característica talvez seja a de maior robustez do framework, pois permite que o sistema construído com ele funcione de forma transparente em relação às características dos dados aos quais gerencia.

Se por acaso houver necessidade de adicionar ou remover dimensões dos dados, não haverá necessidade de modificações nos módulos de processamento de informações, somente os módulos que se comunicam com o banco de dados devem ser atualizados, já que a estrutura das tabelas sofrerá modificações por conta disso.

5. IMPLEMENTAÇÃO DE REFERÊNCIA

Com o propósito de exemplificar como deve ser construído o framework proposto e realizar testes de comportamento em situações reais, uma implementação de exemplo foi desenvolvida usando tecnologias bem conhecidas e amplamente utilizadas na computação.

Os diversos tipos de visualização da informação e a geração instantânea de relatórios previstas no framework são características muito bem exploradas em sistemas web, tais como sites de consultas e portais educativos, por esse motivo a implementação de exemplo trata-se de um sistema web de consulta dos indicadores socioeconômicos dos municípios maranhenses.

As visões foram construídas em HTML e PHP, a arquitetura MVC e os modelos ORM foram implementadas utilizando-se do framework Laravel 3 (LARAVEL, 2013)[10] para desenvolvimento de sistemas web em PHP e os dados do sistema foram armazenados em um banco de dados MySQL.

5.1. Telas

O sistema possui três tipos principais de telas (visões):

- Telas de listagem dos indicadores disponíveis.

  Nesse tipo de tela são listados todos os indicadores cadastrados no sistema e é possível escolher um deles para geração instantânea de um relatório com os dados do indicador escolhido. Há duas telas desse tipo, uma para os indicadores representados pelas variáveis socioeconômicas cujos valores estão realmente cadastrados no banco de dados e a outra para os indicadores frutos do processo de mineração de dados, ou seja, representados pelas variáveis mineradas cadastradas, como se pode ver na Figura 5.1.1.
- Telas de relatório tabular.

  São telas em que o relatório do indicador escolhido é exibido em forma de tabela (Figura 5.1.2). Além dos dados do indicador, o tempo de geração do relatório e a unidade de medida do dado também são mostrados. No caso do indicador escolhido ser representado por uma variável minerada é exibido também o tempo de geração de

[10] <https://github.com/laravel/laravel/tree/v3.0.0>

cada relatório necessário para a construção do indicador e a função de mineração executada. Nesse tipo de tela é possível ainda filtrar o resultado por município e exibi-lo em forma de gráfico (Figura 5.1.3).

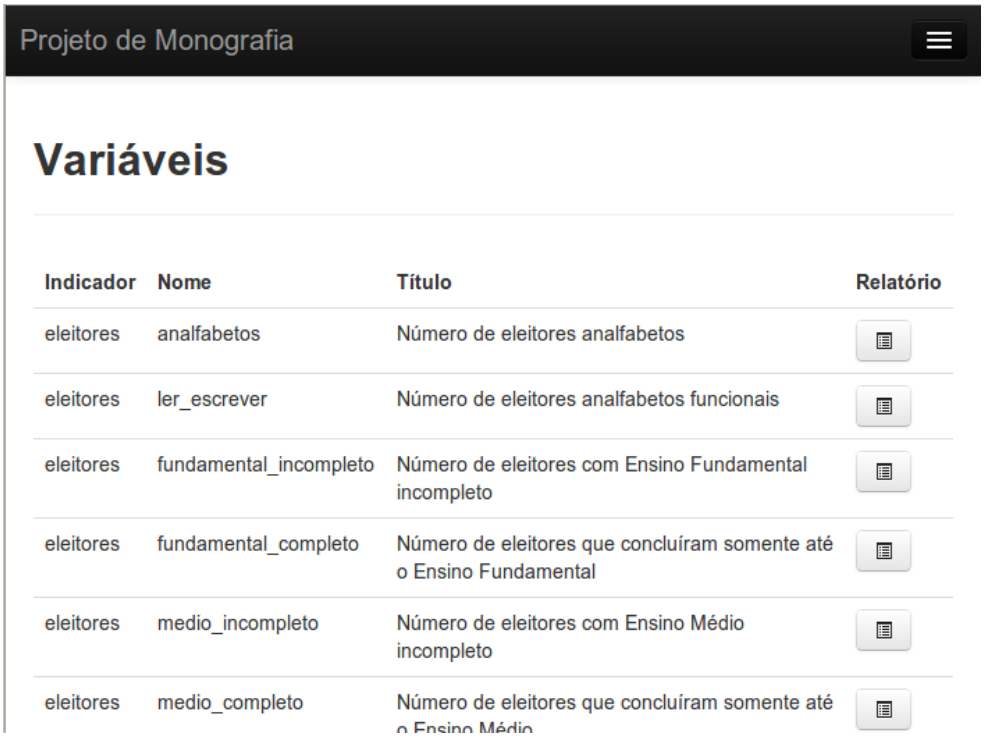

Figura 5.1.1: Tela de listagem dos indicadores cadastrados no framework

Figura 5.1.2: Relatório tabular de um indicador cadastrado no framework

- Telas de relatório gráfico.

  Essa visualização é ideal para comparar indicadores de um grupo pequeno de localidades. O filtro para essa visão é definido na tela de relatório tabular e então os dados dos municípios selecionados são exibidos em um gráfico de linhas.

O indicador escolhido para a visualização sob a forma de relatório é definido por sua chave primária e essa é passada para o controlador da aplicação sob a forma do método HTTP GET via hiperlink, já os filtros para o relatório gráfico são passados para o controlador sob HTTP POST via submissão de formulário.

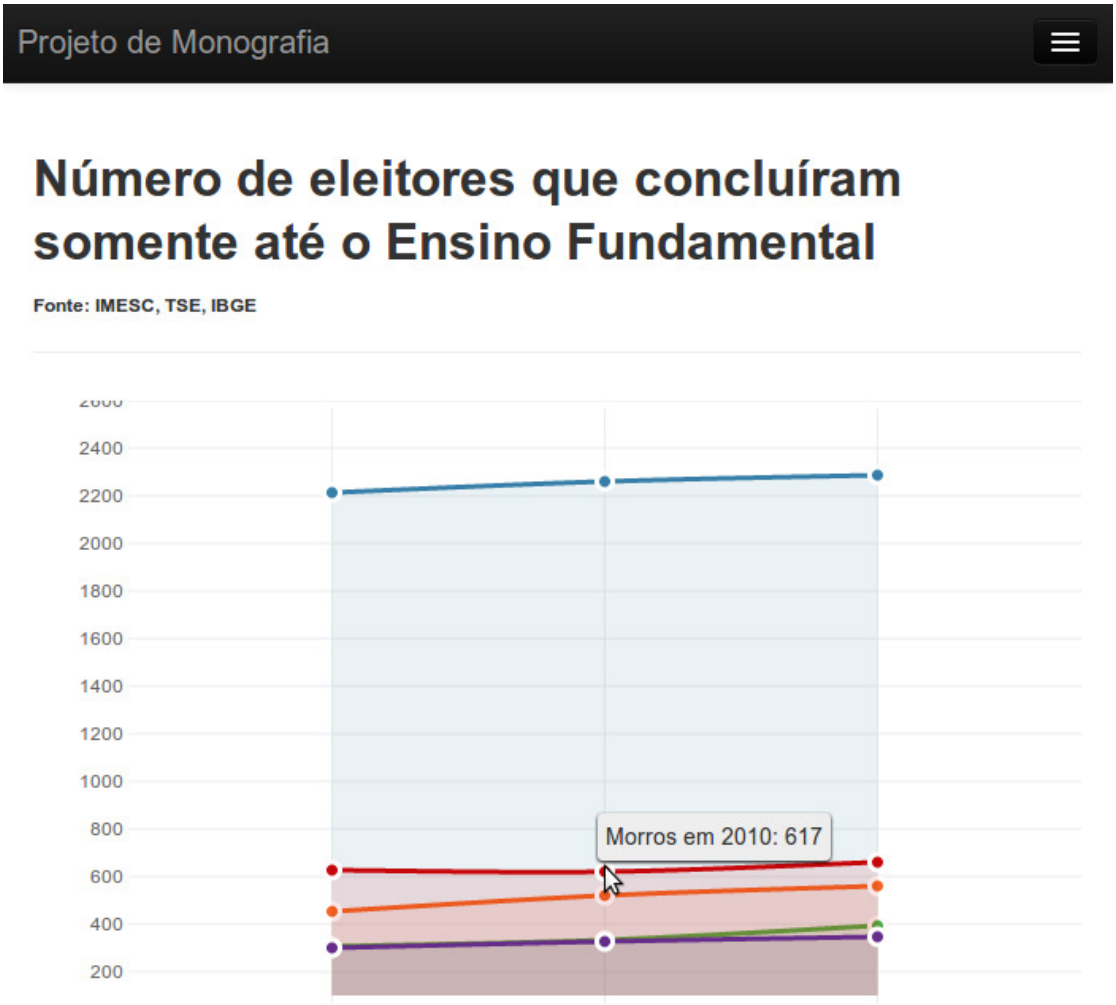

Figura 5.1.3: Relatório gráfico de um indicador cadastrado no framework

Como as telas permitem a análise do tempo de geração dos relatórios, pôde-se perceber que o framework comportou-se bem em relação à característica de geração instantânea. Para variáveis não mineradas o tempo de geração é o menor possível e é quase totalmente representado pelo tempo de query do banco de dados. O SGBD MySQL foi capaz

de proporcionar ao framework um tempo médio de 30 milissegundos para a geração desse tipo de relatório. Já relatórios minerados tem um acréscimo de tempo devido à aplicação da função de mineração, porém nos testes, o pior resultado não ultrapassou os 400 milissegundos, sendo que este envolvia a consulta e o processamento de mais de 6000 estrelas (dados em esquema de estrela) e nove relatórios de variáveis fonte.

De forma geral o tempo de obtenção do relatório cresceu proporcionalmente à quantidade de dados necessários para exibi-lo e o tempo médio foi de 0,05 milissegundos por estrela. O tempo de execução da função de mineração nos relatórios minerados variou dependendo da complexidade das operações envolvidas.

5.2. Classes controladoras

Apenas duas classes controladoras principais fizeram-se necessárias na implementação de exemplo: uma para tratar as requisições referentes às variáveis comuns (indicadores não minerados) e outra, para as variáveis obtidas através do processo de mineração de dados, Variaveis_Controller e Mineracao_Controller, respectivamente.

São controladores muito similares, contendo exatamente as mesmas funções, diferenciando-se apenas no contexto ao qual se aplicam. A listagem dos indicadores cadastrados é tratada pela função action_index que é executada quando o usuário acessa o diretório inicial. A geração de um relatório é iniciada pelo método action_tabular que recebe como parâmetro a chave primária do indicador escolhido pelo usuário, sendo assim, esse método em Variaveis_Controller necessita do nome da tabela de indicadores e da coluna onde se encontra a variável, mas em Mineracao_Controller necessita do id do indicador minerado. O método action_grafico é similar ao action_tabular já que também gera relatório, que neste caso, é para uma visão em gráfico de linhas. Além dos parâmetros de chave primária, action_grafico também recebe os filtros de região via HTTP POST.

Cada método da classe controladora captura e instancia os dados necessários para a construção da visão correspondente. Os métodos de geração de relatório instanciam o modelo ORM da variável requerida e o modelo de relatório correspondente, então o retorno do método make é repassado à visão de relatório apropriada.

```
public function action_index()
{ ... }

public function action_tabular($table, $column)
{ ... }

public function action_grafico($table, $column)
{
   $regions = Input::get('regions');
   ...
}
```

Figura 5.2: Protótipo das funções de Variaveis_Controller

5.3. Modelos ORM

Laravel 3 implementa um completo sistema ORM para ser utilizado pelo programador. A manipulação dos dados das tabelas, o controle de relacionamentos e as consultas no banco de dados pelas classes ORM Variavel e Variavel_M foram herdadas da classe Eloquent do Laravel.

5.4. Método make

O método de construção dos relatórios das variáveis que não são mineradas restringe-se apenas à execução da query de busca dos dados referentes ao indicador e a montagem da matriz contendo esses dados. Para que a matriz seja completamente preenchida é inserido valor nulo nas células que não possuem dados cadastrados.

```
public static function make($variavel)
{
   $results = DB::table($variavel->tabela)
               ->where_not_null($variavel->coluna)
               ->get( array('ano', 'local', 'fonte',
                      $variavel->coluna ));
   ...
}
```

Figura 5.4.1: Trecho de código da função make que executa a query de busca referente a uma variável simples.

O método make das variáveis mineradas é uma função mais complexa que a anterior, pois seu funcionamento deve realizar:

1. Processamento da string de definição das variáveis fonte e da função de mineração.
2. Instanciar e obter sua matriz de dados de cada uma das variáveis fonte definidas.
3. Aplicar a função de mineração sobre as matrizes obtidas.
4. Gerar a nova matriz com os dados minerados.

5.4.1. **Processamento da string**

O framework não especifica o formato da string de definição das variáveis e da função de mineração para indicadores minerados, como exemplo é sugerido os formatos XML e JSON, entretanto é possível também utilizar um formato qualquer definido pelo próprio programador desde que tal formato seja capaz de representar todos os requisitos definidos pelo framework. Para essa implementação de exemplo utilizou-se um formato de string próprio.

Na string de definição de variáveis qualquer cadeia de caracteres não vazia pode representar um alias para uma variável. O caractere '=' (igual) atribui ao alias em definição a correspondência à tabela e coluna ao qual a variável se encontra. O nome da tabela e coluna devem estar separados por um caractere '.' (ponto). Em resumo a estrutura segue como no exemplo abaixo:

var1 = tabelaX.colunaY
var2 = tabelaZ.colunaW

A string da função de mineração é o corpo da função matemática que será aplicada a cada célula das matrizes. Pode conter os operadores aritméticos tradicionais e constantes, mas quando o valor requerido for de uma variável definida o alias da mesma deve ser precedido pelo caractere '$' (cifrão), como no exemplo a seguir:

Definindo o alias para o indicador de eleitores analfabetos:

ANALFABETOS = eleitores.analfabetos

E o alias para o indicador do total de eleitores:

TOTAL = eleitores.total

A função de mineração para calcular a porcentagem de eleitores analfabetos seria:

$ANALFABETOS / $TOTAL * 100

A operação de deslocamento dimensional é representada pelo operador '[ ]' (colchetes). Na função de mineração, logo após o alias de uma variável, pode-se usar o operador de deslocamento dimensional para mover uma ou mais dimensões para a esquerda (sinal de menos), direita (sinal de mais) ou fixar o valor para a dimensão (sinal de igual). As dimensões disponíveis nessa implementação de exemplo são 'local', 'ano' e 'fonte'.

Exemplos:

Crescimento do número de eleitores analfabetos em relação ao ano de 2008:

$ANALFABETOS - $ANALFABETOS[ano=2008]

Crescimento anual de eleitores analfabetos:

$ANALFABETOS - $ANALFABETOS[ano-1]

Diferença do número de eleitores em relação a São Luís:

$TOTAL = $TOTAL[local="São Luís"]

Também é possível usar o operador ternário '?' para representar funções definidas por partes:

Definindo o alias para o relatório minerado do exemplo da porcentagem de eleitores analfabetos (1 é o id desse relatório na tabela variáveis_m):

P = variáveis_m.1

Podemos definir o relatório chamado "Municípios com mais de 20% de eleitores analfabetos":

$P >20? TRUE : FALSE

Seguindo essa especificação, o algoritmo que processa as strings de definição de variáveis e função de mineração constrói as estruturas de dados necessárias para dar continuidade ao processamento do relatório minerado.

5.4.2. **Obtendo as matrizes de dados das variáveis fonte**

Após o processamento da string que define as variáveis fonte necessárias para o processo de mineração, a obtenção da matriz de dados do indicador correspondente é obtido usando um desses dois métodos: se a variável for do tipo simples então a matriz de dados é gerada através do método make da variável, que nesse ponto do código, está instanciada como um modelo ORM; caso a variável fonte seja outra variável minerada então a função executa

procedimento recursivo para obter a matriz de dados.

A característica recursiva do método make das variáveis mineradas permite a geração de relatórios complexos calculados por partes, ou seja, relatórios que servem de variáveis fonte para outros relatórios, porém é importante estar atento para não definir nenhum loop infinito de recursividade.

5.4.3. **Gerando a matriz com os dados minerados**

Os valores das dimensões da matriz gerada são definidos pelo conjunto união dos valores das dimensões das matrizes de variáveis fonte. Aplicando a função de mineração definida sobre cada célula correspondente gera-se o valor que será inserido na matriz de dados minerados. Se não houver um valor correspondente para uma célula da matriz gerada, esta é preenchida com valor nulo.

6. CONCLUSÃO

O trabalho apresentado teve como objetivo propor um framework para auxiliar e agilizar o desenvolvimento de sistemas de consulta de dados socioeconômicos. Para tal feito foi necessário um estudo aprofundado das possíveis estruturas para armazenagem e manipulação de dados multidimensionais, o desenvolvimento de métodos genéricos para extração da informação dessas estruturas, a criação de uma estratégia de alta performance para a mineração de dados socioeconômicos a partir do cadastramento de funções mineradoras e a elaboração de uma interface de programação simples, porém robusta, para o desenvolvimento de softwares a partir do framework.

Os sistemas de análise socioeconômica nacionais, abertos ao público, revelaram-se, por vezes, carentes de funcionalidades de busca ou representação da informação ao usuário. Esse fator aliado com a falta de um framework específico disponível em mercado para o desenvolvimento de sistemas dessa natureza, ao qual o desenvolvedor possa adaptar segundo as necessidades da organização, fazem com que esse trabalho possa ser considerado inovador nessa categoria à medida que outros desenvolvedores possam contribuir com a implementação tomando-a como base e podendo estendê-la.

O sistema de referência desenvolvido para este trabalho atendeu os objetivos para o qual o framework é proposto, funcionando de maneira ágil e estável sobre bases de dados de pequeno e médio porte, porém o framework foi projetado prevendo o caso de utilização de bases de dados muito grandes e esse teste, por conta da indisponibilidade de uma base de dados socioeconômica desse tamanho, não pôde ser realizado.

O framework possui alguns fatores limitantes: a fragmentação interna dos dados, causado pelo agrupamento das variáveis socioeconômicas (estrelas) em uma mesma tabela de tema comum (constelação), que gera desperdício de espaço quando há valores nulos e que, em grande quantidade, pode ocasionar perda de performance; e o processo de mineração que se dá de forma semiautomática, necessitando da intervenção do administrador do sistema para cadastrar explicitamente o algoritmo de mineração que será usado.

Como trabalhos futuros podem ser resolvidos os fatores limitantes do framework: desenvolver uma estratégia que remova ou reduza a fragmentação interna sem diminuir a performance das consultas, principalmente das variáveis de mineração; e, similarmente como é feito em outros sistemas de mineração de dados, incluir métodos para a resolução dos problemas de mineração mais comuns, juntamente com técnicas de inteligência artificial e

aprendizado de máquina, para permitir a análise computacional e estatística dos dados e a descoberta de padrões para o cadastramento automático de variáveis mineradas pelo framework.

7. REFERÊNCIAS BIBLIOGRÁFICAS

(BRAGA, 2005) BRAGA, LUIS PAULO VIEIRA; Introdução à Mineração de Dados. 2a. ed. – Rio de Janeiro: E-papers, 2005.

(CÂMARA; DAVIS, 2001) DAVIS, C.; CÂMARA, G. Arquitetura de Sistemas de Informação Geográfica. In: Câmara, G.; Davis, C.; Monteiro, A. M. V. (Org.). Introdução à ciência da geoinformação. São José dos Campos: INPE, out. 2001. cap. 3.

(CÂMARA ET AL, 1996) CÂMARA, G.; CASANOVA, M.A.; HEMERLY, A.; MEDIEROS, C.M.B.M.; MAGALHÃES, G. C. Anatomia de Sistemas de Informação Geográfica. UNICAMP IX Escola de Computação, 1996 (1a. ed.). SAGRES Editora, Curitiba, 1997 (2a. ed.).

(CAMILO; SILVA, 2009) CAMILO, C.O; SILVA, J.C. Mineração de Dados: Conceitos, tarefas, métodos e ferramentas, Universidade Federal de Goiás (UFC), 2009.

(CHILDINFO, 2013) Childinfo.org. Disponível em: http://www.childinfo.org. Acessado em outubro de 2013.

(DIRK, 2000) RIEHLE DIRK. Framework Design: A Role Modeling Approach. 2000. Tese (PhD). Zurique, Swiss Federal Institute of Technology Zurich, 2000.

(DOCFORGE, 2013) Framework – Docforce. Disponível em: http://docforge.com/wiki/framework. Acessado em: novembro de 2013.

(HAN; KAMBER, 2006) HAN, J; KAMBER, M. Data Mining: Concepts and Techniques. 2a. ed. Elsevier, 2006.

(IBGE, 2013) Sistema IBGE de Recuperação Automática – SIDRA. Disponível em: http://www.ibge.gov.br/home/disseminacao/eventos/workshop/sidra.shtm. Acessado em setembro de 2013.

(IBM, 2013) IBM IPSS Software. Disponível em: http://ibm.com/software/analytics/spss. Acessado em novembro de 2013.

(LARAVEL, 2013) Laravel / A PHP Framework For Web Artisans. Disponível em: https://github.com/laravel/laravel/tree/v3.0.0. Acessado em julho de 2013.

(LONGLEY ET AL, 2005) PAUL A. LONGLEY, MICHAEL F. GOODCHILD, DAVID J. MAGUIRE, DAVID W. RHIND. Systems, Science and Study In: Geographic Information Systems and Science. 2a. ed. Wiley, 2005.

(MINETTO, 2007) MINETTO, ELTON LUIS. Frameworks para Desenvolvimento em PHP. Novatec, 2007.

(NISBET; ELDER; MINER, 2009) NISBET, R; ELDER, J; MINER, G. Statistical Analysis and Data Mining Applications. 1a. ed. Elsevier, 2009.

(Nossa São Luís, 2013) Observatório Nossa São Luís. Disponível em http://www.nossasaoluis.org.br/indicadores2012/apresentacao.html. Acessado em outubro de 2013.

(ORACLE, 2013) Oracle Data Mining. Disponível em: http://oracle.com/technetwork/database/options/advanced-analytics/odm. Acessado em outubro de 2013.

(Portal ODM, 2013) Portal ODM » Sobre. Disponível em: http://www.portalodm.com.br/sobre. Acessado em: outubro de 2013.

(ROHDEN, 2009) ROHDEN, RAFAEL B. Banco de Dados: Relacional X Multidimensional. Disponível em: http://pt.scribd.com/doc/22742853/Artigo-Banco-de-Dados-Relacional-vs-Multidimensional. Acessado em setembro de 2013.

(SAS, 2013) Predictive Analytics and Data Mining Software | SAS. Disponível em: http://www.sas.com/technologies/analytics/datamining. Acessado em novembro de 2013.

(WEKA, 2013) Weka 3 - Data Mining with Open Source Machine Learning Software in Java. Disponível em: http://www.cs.waikato.ac.nz/ml/weka. Acessado em outubro de 2013.

(WITTEN; FRANK, 2005) IAN H. WITTEN, EIBE FRANK. Data Mining Practical Machine Learning Tools and Techniques. 2a. ed. Elsevier, 2005.