



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

TIAGO EDUARDO DA SILVA

**UM *FRAMEWORK* PARA A ANÁLISE ESPACIAL DE FONTES DE DADOS
MULTIDIMENSIONAIS**

**CAMPINA GRANDE – PB
2013**

TIAGO EDUARDO DA SILVA

**UM *FRAMEWORK* PARA A ANÁLISE ESPACIAL DE FONTES DE DADOS
MULTIDIMENSIONAIS**

Dissertação submetida à Coordenação do Curso de Pós-graduação em Ciência da Computação da Universidade Federal de Campina Grande – Campus I – como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Cláudio de Souza Baptista, Ph.D.

Área de concentração: Ciência da Computação.

**CAMPINA GRANDE – PB
2013**



FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCCG

S586f

Silva, Tiago Eduardo da.

Um *framework* para a análise espacial de fontes de dados multidimensionais/ Tiago Eduardo da Silva. – 2013.

123 f. : il. Color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

"Orientação: Prof. Cláudio de Souza Baptista, Ph.D."

Referências.

1. *Geospatial Bussiness Intelligence* (GeoBI). 2. *Spatial On-Line Analytical Processing* (SOLAP). 3. Sistema de Informação Geográfica (SIG). 4. Análise Visual. I. Baptista, Cláudio de Souza . II. Título.

CDU 004.65(043)


**"UM FRAMEWORK PARA A ANÁLISE ESPACIAL DE FONTES DE DADOS
MULTIDIMENSIONAIS"**

TIAGO EDUARDO DA SILVA

DISSERTAÇÃO APROVADA EM 30/08/2013


CLÁUDIO DE SOUZA BAPTISTA, PhD., UFCG
Orientador(a)


ROBSON DO NASCIMENTO FIDALGO, DR., UFPE
Examinador(a)


CARLOS EDUARDO SANTOS PIRES, Dr., UFCG
Examinador(a)

CAMPINA GRANDE - PB

AGRADECIMENTOS

A Deus por me proporcionar mais essa conquista, me provendo de sabedoria e forças para concluir este trabalho de dissertação.

Aos meus pais, Eronides Afonso da Silva e Maria José Conceição da Silva, e aos meus irmãos, Everton Fabiano da Silva e Nivia Fabiana da Silva, por todo o amor, dedicação, apoio e motivação aplicados à minha formação e à conquista dos meus sonhos.

À minha Família.

Ao meu amigo e orientador, professor Dr. Cláudio de Souza Baptista, por toda paciência e dedicação aplicada à minha formação acadêmica e à sua esposa, Dra. Rosilene Santos Baptista, pelas palavras sábias, gentis e motivadoras.

À banca examinadora, professores Dr. Carlos Eduardo Santos Pires e Dr. Robson do Nascimento Fidalgo.

A José Amilton Moura Acioli Filho, pela paciência, esforço e colaboração na realização deste trabalho.

Aos meus amigos-irmãos, Adriano Araújo Santos e Francisco Fernando Roberto Pereira.

Aos meus amigos Hugo Feitosa de Figueirêdo, Damião Ribeiro de Almeida, Maxwell Guimarães de Oliveira, Ana Gabrielle Ramos Falcão, Daniel Farias Batista Leite e Júlio Henrique Rocha.

A todas as pessoas do Laboratório de Sistemas de Informação (LSI) da UFCG, pelos agradáveis momentos de descontração.

A todas as pessoas que participaram e ajudaram, diretamente ou indiretamente, nessa conquista e que não tiveram o nome citado nestes agradecimentos.

RESUMO

As tecnologias de *Business Intelligence* (BI) e de Sistemas de Informações Geográficas (SIG) têm sido utilizadas pelas organizações para extrair informações de diversas fontes de dados, com a finalidade de melhorar o processo de tomada de decisão. Várias soluções visando integrar essas duas tecnologias foram propostas nos últimos anos, quando surgiu o termo *Spatial OLAP*, que passou a ser utilizado para classificar as ferramentas capazes de realizar análises em dados espaciais e convencionais organizados segundo a abordagem multidimensional. Porém, nenhum consenso foi alcançado a respeito da melhor forma de realizar essa integração, o que dificulta a análise de cubos espaciais provenientes de variadas fontes de dados multidimensionais em um mesmo ambiente. Neste trabalho, realiza-se um estudo sobre as principais características do estado da arte em SOLAP e se propõe um ambiente que possibilita a análise dos cubos espaciais provenientes de diversos servidores de dados multidimensionais. Esse objetivo é alcançado com a definição e a implementação de um *framework* SOLAP capaz de analisar e acessar cubos de várias fontes. O *framework* proposto também dispõe de uma linguagem de consulta visual para a análise espacial. Um exemplo prático destinado a analisar as prestações de contas do TCE-AC (Tribunal de Contas do Estado do Acre) foi realizado, visando validar o *framework* que também foi estendido para acessar cubos provenientes das soluções *Microsoft SQL Server Analysis Services* (SSAS) e GeoMondrian.

Palavras-chave: *Geospatial Business Intelligence* (GeoBI), *Spatial On-Line Analytical Processing* (SOLAP), *Sistema de Informação Geográfica* (SIG), Análise Visual.

ABSTRACT

Business Intelligence (BI) and Geographic Information Systems (GIS) technologies have been used by several organizations to extract information from multiple data sources in order to improve the process of decision making. Several solutions for the integration of these two technologies have been proposed in recent years. The term Spatial OLAP arose from these attempts of integration and is now used to classify the tools to perform analysis on spatial and conventional data organized according to the multidimensional approach. Nevertheless, no consensus has been reached regarding the best way to accomplish such integration, making it difficult to perform analysis with spatial cubes from various multidimensional data sources in the same environment. In this work, we realize a study on the main characteristics of the state of the art in SOLAP, and we propose an environment that enables the analysis of spatial cubes from different multidimensional data servers. The proposed framework also has a visual query language for the spatial analysis. To validate the proposed environment, a practical example is conducted and applied to accountability processes of the Court of Auditors of the Acre's State, in Brazil. To perform such case study, the framework was extended to access cubes from Microsoft SQL Server Analysis Services (SSAS) and GeoMondrian.

Key words: Geospatial Business Intelligence (GeoBI), Spatial On-Line Analytical Processing (SOLAP), Geographic Information System (GIS), Visual Analysis.

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
DAO	<i>Data Access Object</i>
DE-9IM	<i>Dimension Extended Nine-Intersection Model</i>
DW	<i>Data Warehouse</i>
ETL	<i>Extract-Transform-Load</i>
GeoBI	<i>Geospatial Business Intelligence</i>
GIS	Geographic Information Systems
GML	<i>Geography Markup Language</i>
JDBC	<i>Java Database Connectivity</i>
JPA	<i>Java Persistence API</i>
JSF	<i>JavaServer Faces</i>
JSON	<i>JavaScript Object Notation</i>
JTS	<i>Java Topology Suite</i>
MDX	<i>Multidimensional Expressions</i>
OGC	<i>Open Geospatial Consortium</i>
OLAP	<i>On-line Analytical Processing</i>
OLTP	<i>Online Transaction Processing</i>
ORDBMS	<i>Object Relational Database Management System</i>
PL/SQL	<i>Procedural Language/Structured Query Language</i>
ROLAP	<i>Relational OLAP</i>
SDW	<i>Spatial Data Warehouse</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SIG	Sistema de Informação Geográfica
SOLAP	<i>Spatial OLAP</i>
SQL	<i>Structured Query Language</i>
SSAS	<i>SQL Server Analysis Services</i>
TCE-AC	Tribunal de Contas do Estado do Acre
TCU	Tribunal de Contas da União
TI	Tecnologia da Informação
WFS	<i>Web Feature Service</i>
WKT	<i>Well-Known Text</i>
WMS	<i>Web Map Service</i>
WPS	<i>Web Processing Service</i>
XML	<i>Extensible Markup Language</i>
XMLA	<i>XML for Analysis</i>

LISTA DE FIGURAS

Figura 1 – Arquitetura típica de BI - Extraído de [1].....	21
Figura 2 – Os três tipos de dimensões espaciais de acordo com Bédard et al. Extraído de [21].....	25
Figura 3 – Aplicação da Álgebra Tabular na ferramenta Tableau	29
Figura 4 – Interface da ferramenta <i>JMAP Spatial OLAP</i>	32
Figura 5 – Interface da ferramenta GeWolap	33
Figura 6 – Interface da ferramenta SOVAT	34
Figura 7 – Interface desktop da ferramenta PIET	35
Figura 8 – Interface desktop da ferramenta Golapware	36
Figura 9 – Metamodelo do Mapwarehouse. Extraído de [3].....	38
Figura 10 - Diagrama de classes para a Consulta Conceitual. Extraído de [3]	38
Figura 11 – Interface da ferramenta Mapwarehouse.....	39
Figura 12 – Interface da ferramenta Polaris	41
Figura 13 – Interface da ferramenta Tableau	41
Figura 14 – Interface da extensão da ferramenta SAIKU para o projeto Desourb.....	42
Figura 15 – Diagrama de classes para o cubo espacial.	46
Figura 16 – Divisória Camadas para a análise espacial	48
Figura 17 – Arquitetura da solução SOLAP	51
Figura 18 – Interface para a especificação visual de uma consulta SOLAP	53
Figura 19 – Dimensão DIM TEMPO.....	53
Figura 20 – Exemplo de tabela gerada pelas divisórias Colunas e Linhas.....	55
Figura 21 – Componente para a navegação nas páginas.....	55
Figura 22 – Exemplo de tela para adição de um campo à divisória “Filtros”	56
Figura 23 – Divisórias para a especificação da formatação da tabela.....	57
Figura 24 – Filtro convencional para os membros.....	57
Figura 25 – Filtro geográfico para os membros	58
Figura 26 – Divisória “Camadas”	58
Figura 27 – Componente para a escolha do tipo de painel.....	59
Figura 28 – Especificação visual para o tipo de marcação Gráfico	60
Figura 29 – Especificação visual para o tipo de marcação Legenda.....	60
Figura 30 – Tabela marcada com texto	61
Figura 31 – Tabela marcada com gráfico resultante da especificação ilustrada na Figura 28	62
Figura 32 – Tabela marcada com legenda resultante da especificação ilustrada na Figura 29	62
Figura 33 – Menu para a geocodificação	64
Figura 34 – Aba para a seleção do nível para o processo de geocodificação.....	65
Figura 35 – Prévia dos valores da propriedade dos membros.....	66
Figura 36 – Aba para a seleção da tabela espacial	66
Figura 37 – Aba com o resultado do processo de geocodificação	67
Figura 38 – Tela listar níveis geocodificados.....	68
Figura 39 – Tela editar mapeamento da geocodificação	68
Figura 40 – Fachada para o engenho SOLAP	69
Figura 41 – Tela de gerenciamento dos cubos	71

Figura 42 – Diagrama de atividades para a realização de consultas na solução proposta.....	71
Figura 43 – Diagrama de classe para a consulta visual.....	72
Figura 44 – Diagrama de classe para o resultado da consulta visual.....	73
Figura 45 – Cubo para os exemplos.....	74
Figura 46 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas”.....	74
Figura 47 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas” e o nível de medidas no eixo “Colunas”.....	75
Figura 48 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas”, o nível de medidas no eixo “Colunas” e os níveis “Ano” e “Mês do Ano” no eixo “Páginas”.....	76
Figura 49 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas”, o nível de medidas no eixo “Colunas” e os níveis “País” e “Estado” no eixo “Camadas”.....	78
Figura 50 – <i>Template</i> do componente para informar as propriedades da conexão.....	81
Figura 51 – Diagrama de classes para os objetos LoaderPopup e LoaderBean.....	82
Figura 52 – Classe JDBCDataSource.....	83
Figura 53 – Diagrama de classes para o GenericDAO.....	83
Figura 54 – <i>Popup</i> para conexão à fonte de dados.....	84
Figura 55 – Tela para a escolha do cubo.....	85
Figura 56 – Diagrama de classes para a classe AbstractOlap4jXMLACubeConverter.....	86
Figura 57 – Diagrama de Classes para os DAOs do engenho XMLA.....	87
Figura 58 – Modelo do DW utilizado para construir o cubo Empenho.....	93
Figura 59 – Modelo do DW utilizado para o cubo Liquidação.....	93
Figura 60 – Modelo do DW para o cubo Pagamento.....	94
Figura 61 – Resultado da consulta 1 - nível de detalhe para o estado do Acre.....	95
Figura 62 – Resultado da Consulta 1 - nível de detalhe para as mesorregiões do Acre.....	96
Figura 63 – Resultado da Consulta 1 - nível de detalhe para as microrregiões do Acre.....	96
Figura 64 – Resultado da consulta 1 - nível de detalhe para os municípios do Acre.....	97
Figura 65 – Resultado da consulta 2.....	98
Figura 66 – Filtro convencional para as funções.....	99
Figura 67 – Filtro espacial para os municípios.....	99
Figura 68 – Resultado do filtro espacial refletido no filtro convencional.....	100
Figura 69 – Resultado da consulta 3.....	100
Figura 70 – Filtro para os municípios contidos nas microrregiões Sena Madureira ou Tarauacá.....	101
Figura 71 – Resultado da consulta 4.....	102
Figura 72 – Resultado da consulta 5.....	103
Figura 73 - Consulta visual com os níveis “DESCRIÇÃO TIPO EMEPNHO” e “DESCRIÇÃO SUBFUNÇÃO” no eixo “Linhas”, o nível de medidas no eixo “Colunas”, o nível “ANO” no eixo “Filtros” e os níveis “NOME MUNICÍPIO” e “NOME MICRORREGIÃO” no eixo “Camadas”....	114

LISTA DE TABELAS

Tabela 1 – Quadro comparativo entre as soluções investigadas	44
Tabela 2 – Resultado consulta visual	76
Tabela 3 – Novo quadro comparativo	88

LISTA DE CÓDIGOS FONTE

Código 1 – JSON com as tuplas para o eixo "Linhas"	75
Código 2 – Código JSON contendo o resultado do exemplo ilustrado na Figura 47.....	75
Código 3 – Código JSON contendo o resultado do exemplo ilustrado na Figura 48.....	77
Código 4 – Código JSON contendo o resultado do exemplo ilustrado na Figura 49.....	78
Código 5 – Pseudo-código para a tradução da consulta visual para a linguagem de consulta MDX..	112
Código 6 – Consultas MDX geradas pelo pseudo-código ilustrado no Código 5 para a consulta visual ilustrada na Figura 73.....	114

SUMÁRIO

1. INTRODUÇÃO	15
1.1. PROBLEMÁTICA.....	16
1.2. OBJETIVOS	17
1.3. RELEVÂNCIA E CONTRIBUIÇÕES DO ESTUDO	17
1.4. ESTRUTURA DA DISSERTAÇÃO	18
2. FUNDAMENTAÇÃO TEÓRICA	19
2.1. <i>GEOSPATIAL BUSINESS INTELLIGENCE</i> (GEOBI).....	19
2.2. DATA WAREHOUSE (DW)	20
2.3. ON-LINE ANALYTICAL PROCESSING (OLAP)	22
2.4. <i>SPATIAL DW</i> (SDW).....	24
2.5. <i>SPATIAL OLAP</i> (SOLAP).....	24
2.5.1. Dimensão espacial.....	25
2.5.2. Medida espacial.....	26
2.6. LINGUAGEM FORMAL VIZQL	27
2.7. CONSIDERAÇÕES.....	30
3. TRABALHOS RELACIONADOS.....	31
3.1. JMAP SPATIAL OLAP.....	31
3.2. GeoWOLAP	32
3.3. SOVAT	33
3.4. PIET	34
3.5. GOLAPWARE.....	36
3.6. MAPWAREHOUSE	37
3.7. POLARIS	39
3.8. DESOURB/Saiku.....	42
3.9. CONSIDERAÇÕES.....	43
4. UMA PROPOSTA DE <i>FRAMEWORK</i> PARA A ANÁLISE ESPACIAL DE FONTES DE DADOS MULTIDIMENSIONAIS.....	46
4.1. MODELO PARA CUBOS ESPACIAIS.....	46
4.2. EXTENSÃO DA LINGUAGEM VIZQL	48
4.3. ARQUITETURA	51
4.3.1. Módulo de especificação visual de consultas.....	52
4.3.2. Módulo de visualização dos dados.....	63

4.3.3.	Módulo gerenciador de mapas	63
4.3.4.	Módulo de Acesso ao repositório de dados espaciais	63
4.3.5.	Módulo de geocodificação	64
4.3.6.	Engenho SOLAP.....	69
4.4.	PONTOS DE EXTENSÃO DA SOLUÇÃO	79
4.4.1.	Fachada para o engenho SOLAP.....	79
4.4.2.	Interface para as propriedades de conexão.....	80
4.4.3.	Engenho SOLAP para servidores XMLA	84
4.5.	CONSIDERAÇÕES.....	87
5.	AVALIAÇÃO E EXEMPLO PRÁTICO.....	90
5.1.	EXTENSÕES DO ENGENHO SOLAP PARA SERVIDORES XMLA	90
5.2.	EXEMPLO PRÁTICO APLICADO AO TRIBUNAL DE CONTAS DO ESTADO DO ACRE 91	
5.2.1.	Esquema conceitual.....	91
5.2.2.	Consultas	94
5.3.	CONSIDERAÇÕES.....	104
6.	CONCLUSÕES E TRABALHOS FUTUROS	105
6.1.	CONCLUSÕES.....	105
6.2.	TRABALHOS FUTUROS.....	106
	REFERÊNCIAS.....	108
	Apêndice A.....	112
	Anexo A	116

1. INTRODUÇÃO

Com o crescente volume de dados provenientes das mais diversas fontes, houve um considerável aumento de investimentos em tecnologias de suporte à decisão, denominadas de Inteligência Empresarial ou *Business Intelligence* (BI). As tecnologias de BI são capazes de transformar dados em informações úteis, além de fornecer uma visão histórica, corrente e preditiva das operações de negócio de uma empresa.

Segundo Chaudhuri et al. [1], as empresas estão explorando suas bases de dados por meio de novas técnicas de análise, com o intuito de apoiar a tomada de decisão e de oferecer novas funcionalidades aos seus clientes, fazendo com que a maioria das empresas utilize ferramentas de BI. Dentre as ferramentas de BI mais utilizadas estão as denominadas *On-line Analytical Processing* (OLAP), que permitem a rápida exploração e análise dos dados que são armazenados em vários níveis de agregação segundo a abordagem multidimensional.

Em paralelo a isso, sabe-se que aproximadamente 80% dos dados das corporações possuem um componente espacial [2], o que desperta o interesse das corporações em explorar esses dados com o objetivo de obter novas informações que ajudem os usuários a tomarem as melhores decisões. Sistemas de Informação Geográfica (SIG), ou *Geographic Information Systems* (GIS), vêm sendo utilizados para a descoberta de informações e são capazes de armazenar, recuperar, manipular, gerenciar, analisar e visualizar dados geográficos, também denominados de dados espaciais.

Os dados espaciais são responsáveis por identificar uma região ou superfície da terra e estão relacionados à localização no espaço, ou seja, com as coordenadas geográficas, o que possibilita a visualização desses dados em mapas. A necessidade de extrair informações dos dados espaciais para melhorar o processo de tomada de decisão fez com que surgisse uma nova categoria de ferramentas conhecida como *Geospatial Business Intelligence* (GeoBI), que integra tecnologias de SIG e BI.

O termo *Spatial OLAP* (SOLAP) é utilizado para categorizar as ferramentas de GeoBI que integram as funcionalidades de SIG e OLAP. Essas ferramentas são capazes de realizar análises em dados espaciais e de apresentar os resultados dessas análises por meio de gráficos e de mapas. Assim, a pesquisa desenvolvida nesta dissertação se insere no contexto das soluções SOLAP.

1.1. PROBLEMÁTICA

A necessidade de integrar as tecnologias de SIG e OLAP por parte das corporações fez com que várias soluções fossem propostas e desenvolvidas na Academia, porém nenhum consenso foi alcançado a respeito da melhor forma de realizar essa integração. Segundo Cardoso [3], as soluções divergem quanto aos aspectos de implementação, ao modelo de dados e ao uso de operadores para as consultas. As soluções *JMap Spatial Olap* [4], *GeoWolap* [5], *SOVAT* [6], *Piet* [7], *Golapware* [8], *Mapwarehouse* [3], entre outras, visaram integrar dados espaciais ao ambiente de *Data Warehousing*. Assim, surgiu o termo *Spatial Data Warehouse* (SDW) e, conseqüentemente, o *Spatial OLAP* (SOLAP).

Ainda referente à integração dos sistemas SIG e BI, trabalhos investigaram e propuseram formas de disponibilizar os dados na Web para garantir a troca desses dados e a interoperabilidade. Silva et al. [5] propuseram um *Web Service* denominado GMLA WS, que combina XMLA, GML e WFS. Dubé et al. [9] apresentaram um formato XML para disponibilizar e trocar cubos SOLAP via *Web Service*. Em 2011, o *Open Geospatial Consortium* (OGC) [10] publicou um relatório (*white paper*) [11] contendo uma avaliação sobre a forma como os padrões OGC (e.g. WMS, WFS, WPS, etc.) podem ser estendidos, com o intuito de intensificar o uso das informações geoespaciais e a interoperabilidade das aplicações de GeoBI. Porém, até então, nenhuma extensão ou padrão foi publicado pelo OGC nesse sentido.

No estudo realizado nesta pesquisa, observaram-se algumas características determinantes no desenvolvimento de soluções SOLAP:

- I. Permitir a criação de consultas por meio de uma linguagem de especificação visual;
- II. Possuir uma visualização integrada dos dados multidimensionais e espaciais;
- III. Possuir uma interface Web;
- IV. Permitir a comparação dos dados;
- V. Dar suporte aos operadores espaciais;
- VI. Utilizar tecnologias abertas;
- VII. Permitir a visualização dos dados por meio de mapas, tabelas e gráficos;
- VIII. Ser extensível.

Além dessas características encontradas na literatura, elencamos neste trabalho novas características que completam às mencionadas anteriormente, a saber:

- IX. Permitir o acesso a várias fontes de dados multidimensionais (servidores de cubos);
- X. Possibilitar a geocodificação dos dados.

No capítulo 3, discutiremos a necessidade de tais características.

A falta de consenso para a integração SIG-OLAP e de padrões para a disponibilização dos dados espaciais e multidimensionais dificulta a utilização de variadas fontes de dados para a extração de informações úteis para melhorar o processo de tomada de decisão das corporações. Diante do exposto, nesta dissertação, propõe-se um *framework* para a análise espacial que possibilita a utilização de várias fontes multidimensionais, com as características I, II, III, IV, V, VI, VII, VIII, IX e X.

1.2. OBJETIVOS

O objetivo deste trabalho é de propor um *framework* SOLAP que possibilite a análise de dados espaciais multidimensionais provenientes de diversas fontes de dados.

Para isso, foram elencados os seguintes objetivos específicos:

- Propor uma interface gráfica que permita a análise dos dados espaciais multidimensionais;
- Propor um *framework* SOLAP capaz de acessar várias fontes de dados espaciais multidimensionais;
- Desenvolver um cliente SOLAP que implemente as características do *framework* proposto, a fim de facilitar sua validação;
- Validar o *framework* por meio de um exemplo prático.

1.3. RELEVÂNCIA E CONTRIBUIÇÕES DO ESTUDO

A principal contribuição deste trabalho está em definir um *framework* capaz de acessar várias fontes de dados espaciais multidimensionais, partindo de características observadas nas principais abordagens existentes e preenchendo lacunas identificadas nessas abordagens.

A utilização de operadores espaciais, o acesso a várias fontes de dados espaciais multidimensionais e a utilização de uma linguagem de consulta visual são as principais características desse *framework*.

1.4. ESTRUTURA DA DISSERTAÇÃO

Os próximos capítulos estão organizados da seguinte forma:

- Capítulo 2: Apresenta a fundamentação teórica, ou seja, os conceitos básicos que conduzem ao entendimento do tema em estudo;
- Capítulo 3: Os trabalhos relacionados ao tema SOLAP são apresentados e comparados com a solução proposta;
- Capítulo 4: Apresenta o framework proposto para a análise espacial de fontes de dados espaciais multidimensionais, a arquitetura e alguns detalhes de implementação;
- Capítulo 5: Traz um estudo de caso no contexto do Tribunal de Contas do Estado do Acre (TCE-AC) e aspectos de implementação de duas extensões do *framework* utilizados para a validação do trabalho;
- Capítulo 6: Traz as conclusões a respeito do tema pesquisado e sugestões de trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os pressupostos teóricos que fundamentam esta dissertação, em que são abordados os seguintes conteúdos: *Geospatial Business Intelligence*, *Data Warehouse*, *On-line Analytical Processing*, *Spatial Data Warehouse*, *Spatial On-Line Analytical Processing* e Linguagem Formal VizQL.

2.1. GEOSPATIAL BUSINESS INTELLIGENCE (GEOBI)

Grandes quantidades de dados são coletadas nas operações diárias das organizações, porém estão armazenados em sistemas otimizados para realizar um grande número de transações e permitir o acesso concorrente, mantendo a consistência dos dados e a tolerância a falhas [12] [13]. A maioria desses sistemas, denominados de *Online Transaction Processing* (OLTP), armazena os dados utilizando o modelo relacional em Sistemas de Gerenciamento de Banco de Dados (SGBD) e os mantém em formatos heterogêneos e na forma normalizada. Isso dificulta a extração de informações para auxiliar os gestores no processo de tomada de decisão.

Os gestores precisam de informações agregadas e sumarizadas, rápidas comparações no espaço e no tempo, síntese de milhões de ocorrências, descoberta de tendências e outras operações complexas para apoiar o processo tático e estratégico de tomada de decisão [4].

No início da década de 70, surgem os primeiros sistemas de apoio à decisão, em contraste com os sistemas transacionais. Ao longo dos anos, várias aplicações de suporte à decisão surgiram e expandiram o domínio de apoio à decisão. No início da década de 90, Howard Dressner, um analista do Gartner Group, cunhou o termo *Business Intelligence*, que hoje é amplamente utilizado para descrever aplicações analíticas [14].

Inteligência Empresarial ou *Business Intelligence* (BI) é uma coleção de tecnologias voltadas para dar suporte à decisão. De acordo com Chaudhuri et al. [1], BI é uma coleção de tecnologias de suporte à decisão destinada a permitir que executivos, gerentes ou analistas possam tomar decisões melhores e mais rápidas. Relatórios, *dashboards*, *scoreboards*, *online analytics processing* (OLAP), mineração de dados, mineração de processos, *business performance management analytics* e análise preditiva são funcionalidades comuns das tecnologias de BI que dão uma visão histórica, corrente e preditiva das operações de negócio de uma empresa [11]. Porém, tecnologias de BI não tiram proveito dos dados espaciais.

Sistemas de Informação Geográfica (*Geographical Information Systems - GISs*) foram projetados para trabalhar com dados referenciados por coordenadas espaciais ou geográficas. Todavia esses sistemas utilizam a abordagem transacional, o que dificulta a análise eficiente desses dados. Embora Sistemas de Informação Geográfica (SIG) e *Business Intelligence* (BI) tenham seguido diferentes caminhos de implementação e desenvolvimento, o aumento da exigência da exploração dos dados espaciais para apoiar o processo de tomada de decisão resultou na formação de um modelo de colaboração dessas duas tecnologias.

Geospatial Business Intelligence (GeoBI) combina a análise espacial e a visualização em mapas, bem como a mineração de dados espaciais, com as tecnologias de BI. A visualização interativa em mapas permite que usuários façam *zoom* e *pan* em áreas de interesse; acessem ferramentas espaciais, como *buffer*, naveguem e filtrem as informações de uma forma que é difícil ou até mesmo impossível em soluções de BI. Com a análise espacial, é possível explorar relações entre os dados e a localização geográfica e identificar tendências que não eram facilmente vistas ou até impossíveis na análise tradicional, como proximidade e coincidência de eventos e ativos.

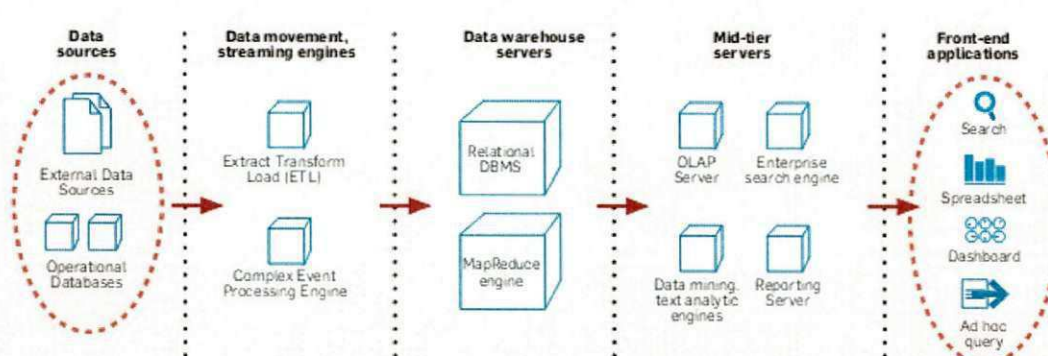
2.2. DATA WAREHOUSE (DW)

Segundo Chaudhuri et al. [1], uma arquitetura típica de BI tem como componente central um *Data Warehouse* (DW), conforme a arquitetura apresentada na Figura 1. Um *Data Warehouse*, ou Armazém de Dados, é um banco de dados multidimensional voltado para a análise estratégica e a tomada de decisão. Hoje, é parte essencial da infraestrutura de TI devido à necessidade das corporações de obterem informações oriundas, principalmente, dos dados coletados das bases operacionais e de fontes de dados externas. O DW é otimizado para facilitar análises complexas e aumentar o desempenho do banco de dados em consultas com milhares de ocorrências.

Inmon [15] refere que “um *Data Warehouse* é uma coleção de dados orientada por temas, integrada, variante no tempo e não-volátil, que dá apoio ao processo de tomada de decisões gerenciais”. Um DW é voltado para a análise de grandes quantidades de dados, diferente das bases de dados operacionais, também conhecidas pelo termo OLTP (*On-line Transaction Processing*), que auxiliam o controle operacional, ou seja, os corriqueiros processos de negócio. Essa diferente abordagem impacta em o que armazenar e como armazenar os dados. Por esse motivo, um DW é projetado em torno de temas (subjects) de interesse das corporações, facilitando a análise dos dados; é integrado, ou seja, os dados de

um DW são oriundos de diferentes bases operacionais que podem armazenar os dados em diferentes formatos; é variante no tempo, pois armazena dados históricos, não mantém apenas os dados mais recentes; é não volátil, porque não permite operações de remoção (*deletes*) nem atualizações (*updates*), apenas carga de dados e consultas; é desnormalizado e promove mais eficiência nas consultas.

Figura 1 – Arquitetura típica de BI - Extraído de [1]



Um subconjunto do DW é o *Data Mart*, que é construído para satisfazer às necessidades de um departamento e pode armazenar apenas um tema (e.g. Vendas). Já um DW é compartilhado com toda a organização e armazena vários temas [16]. A carga de dados de um DW é feita por meio de um processo de extração, transformação e carga, denominado de ETL (*Extraction-Transformation-Load*), em cujo processo são extraídos os dados de várias fontes operacionais; em seguida, os dados são transformados conforme o formato utilizado no DW, as correções são aplicadas aos dados de má qualidade e, por fim, os dados transformados e corrigidos são adicionados ao DW.

Os conceitos de fatos, medidas, dimensões e membros são utilizados para modelar DWs. Os temas que se deseja analisar são denominados de Fatos, por exemplo, o tema Venda é um fato de análise no mercado varejista. Atributos numéricos denominados medidas estão associados aos fatos. As medidas são propriedades que o usuário deseja analisar, por exemplo, a quantidade de itens vendidos e o valor da venda. Os fatos são definidos pela combinação dos valores das dimensões. Assim, as dimensões caracterizam os fatos. Cada instância de uma dimensão é denominada membro da dimensão. Uma combinação dos membros das dimensões identifica, de forma unívoca, um fato.

Um modelo multidimensional pode ser representado de duas formas, em um banco de dados relacional, por meio do esquema estrela ou do esquema floco de neve. No esquema estrela, uma tabela é criada para cada dimensão e para cada fato. A tabela criada para o fato é denominada tabela de fatos e possui uma coluna para cada medida e uma para cada dimensão. Nesse esquema, as tabelas das dimensões relacionam-se diretamente com as tabelas de fatos, formando uma figura semelhante a uma estrela.

A diferença entre o esquema floco de neves e o estrela é de que o primeiro pode representar uma dimensão por meio de várias tabelas, ou seja, as dimensões podem ser normalizadas. Uma figura complexa semelhante a um floco de neve surge quando o número de dimensões normalizadas aumenta.

2.3. ON-LINE ANALYTICAL PROCESSING (OLAP)

De acordo com Rivest et al. [4], para explorar os dados de forma interativa, as soluções de BI mais utilizadas são as ferramentas OLAP. A abordagem OLAP suporta a natureza interativa do processo analítico, porque permite que o usuário explore e navegue por meio de diferentes temas (dimensões), em diferentes níveis de detalhe, e, rapidamente, visualizar os fatos ou dados intersectando as dimensões, qualquer que seja o nível de agregação. Ainda em [4], Rivest et al. citam Caron [17], que define OLAP como “uma categoria de software destinada à rápida exploração e à análise de dados, com base em uma abordagem multidimensional, com vários níveis de agregação”.

As ferramentas OLAP implementam técnicas de análise utilizadas para explorar, principalmente, modelos multidimensionais denominados cubos ou hipercubos. Com elas, gestores podem realizar consultas *ad-hoc*, de forma rápida e interativa, utilizando um conjunto de operações que permitem navegar e explorar os fatos em vários níveis de detalhes e perspectivas.

O conceito de cubo é utilizado pelas ferramentas OLAP e é definido em [16] como sendo uma estrutura multidimensional utilizada para analisar os dados de forma eficiente e responsável por manter as informações das empresas agregadas e sumarizadas. O termo cubo sugere a existência de três dimensões, porém um cubo pode ter qualquer número de dimensão.

Os conceitos dos termos fatos, medidas, dimensões e membros também são utilizados pelas ferramentas OLAP, além de conceitos como células, hierarquias e níveis. Os cubos são compostos por dimensões e medidas. As medidas são propriedades que o usuário deseja

analisar (e.g. quantidade de itens vendidos e valor da venda) e estão associadas a um fato. Uma função de agregação é utilizada para combinar vários valores de uma medida em um único valor.

A intersecção das dimensões de um cubo identifica de forma unívoca as células, ou seja, as células são identificadas pela combinação dos membros das dimensões. Uma célula tem medidas associadas que, geralmente, são valores numéricos.

Uma dimensão pode ser organizada em hierarquias de níveis, cada um dos quais representa um nível de detalhe dos dados. Um exemplo é a dimensão local, com três níveis: Cidade -> Estado -> ALL. O nível ALL representa todos os membros da dimensão. Toda hierarquia tem o nível ALL. Cidade é o nível mais detalhado. Os níveis são compostos por membros que podem ser vistos como instâncias dos níveis. Os membros têm determinado número de propriedades associadas para armazenar informações simples e não hierárquicas (tamanho da população).

Uma arquitetura OLAP comum pode ser dividida em três partes: a) a base de dados, geralmente um DW; b) o servidor OLAP, também denominado fonte de dados multidimensionais ou servidor de cubos, onde estruturas denominadas cubos são mantidas; e c) o cliente OLAP, que acessa os cubos de dados disponibilizados pelo servidor OLAP e permitem que os usuários finais analisem os dados utilizando diferentes métodos de visualização e operadores OLAP.

As operações OLAP permitem que os analistas naveguem e explorem os dados em diferentes níveis de detalhes e perspectivas. As principais operações são *roll-up*, *drill-down*, *slice and dice* e *pivot*. A operação *roll-up* permite navegar de um nível para outro nível superior em uma hierarquia, diminuindo a quantidade de detalhes. A operação *drill-down* funciona de forma inversa, pois, por meio dela, é possível ir de um nível para outro inferior na hierarquia, e isso aumenta o nível de detalhes.

Ao selecionar um subconjunto dos membros de uma dimensão, um subconjunto do cubo é exibido, assim obtém-se um *slice* do cubo. Um *dice* pode ser obtido por meio de restrições aos membros do cubo. O operador responsável por criar esses subconjuntos é denominado *slice and dice*. Selecionar as vendas ocorridas no ano de 2009 ou as vendas ocorridas em Campina Grande no ano de 2008 é um exemplo do uso desse operador. O operador *pivot* permite alterar a perspectiva de visualização do cubo. Quando ocorre um pivoteamento as dimensões são rotacionadas, ou seja, elas mudam de eixo.

A extensão do DW convencional para dar suporte aos dados espaciais denomina-se *Spatial DW* (SDW). Da mesma forma, a extensão das ferramentas OLAP para dados espaciais denomina-se *Spatial OLAP* (SOLAP). As próximas seções detalham essas tecnologias.

2.4. *SPATIAL DW* (SDW)

O *Spatial DW* (SDW) pode ser entendido como a extensão de um DW convencional, com suporte para dados espaciais (georreferenciados). Isso é possível pela adição de medidas e dimensões espaciais mantendo as características tradicionais de um DW.

A integração de dados espaciais em DWs traz grandes benefícios para o processo de tomada de decisão e permite a descoberta de novas informações que, antes, não era possível, devido à ausência desse tipo de dados. Consultas mais abrangentes como “Qual é o total de vendas dos produtos da categoria ração para peixe nas cidades a uma distância de até 3Km de Campina Grande e que são cortados pelo rio Paraíba?” passam a ser possíveis. Porém, ainda não existe um consenso a respeito dos conceitos relacionados aos SDWs. Diferentes modelos abordam diferentes conceitos. Os principais conceitos são apresentados nas seções 2.5.1 e 2.5.2.

2.5. *SPATIAL OLAP* (SOLAP)

Para explorar e analisar dados convencionais e espaciais, surgiu uma nova categoria de ferramentas que combina o poder das ferramentas GIS e OLAP, denominada *Spatial OLAP* (SOLAP).

Rivest et al. [18] definiram o SOLAP como “uma plataforma construída especialmente para suportar a análise espaço-temporal e exploração de dados, seguindo uma abordagem multidimensional composta de níveis de agregação que podem ser visualizados em tabelas, gráficos e mapas”. A arquitetura de um sistema SOLAP é composta de um armazém de dados, que pode ser espacial, um servidor SOLAP e um cliente SOLAP. Os dados espaciais podem ser mantidos em um SDW ou em outro ambiente, onde apontadores irão relacionar os dados convencionais aos dados espaciais. Assim, a integração entre os dados espaciais e as ferramentas OLAP pode ser feita por meio de duas abordagens: a) a federada, em que os dados espaciais não se encontram no mesmo ambiente dos dados convencionais; ou b) a integrada, em que ambos os tipos de dados se encontram no mesmo ambiente (e.g. SDW). O

servidor SOLAP é responsável por manter e disponibilizar os cubos espaciais. Por fim, o cliente SOLAP permite a visualização dos dados em tabelas, gráficos e mapas.

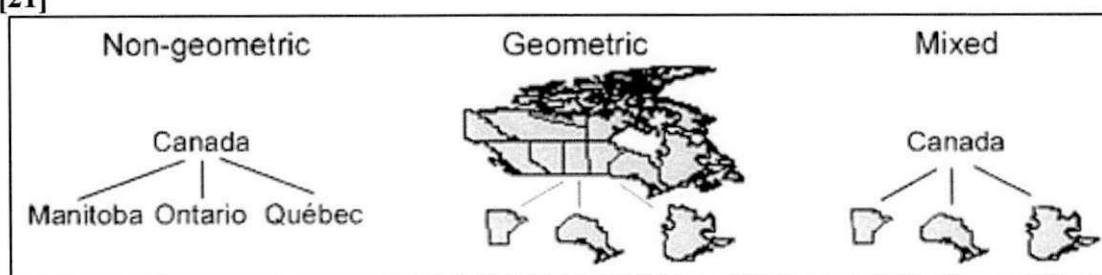
A fim de aproveitar a natureza espacial dos dados envolvidos nas dimensões, operações como *spatial roll-up* e *spatial drill-down* são suportadas pelas ferramentas SOLAP para permitir a navegação nas hierarquias espaciais. Com outros operadores espaciais pode-se selecionar subconjuntos do cubo utilizando restrições topológicas e/ou geométricas.

Nas ferramentas SOLAP, há a extensão de dois conceitos OLAP: dimensão e medida. Esses conceitos serão detalhados nas próximas seções.

2.5.1. Dimensão espacial

Para definir a dimensão espacial, dois pontos de vistas foram considerados: o de Bédard et al. [19] e o de Han et al. [20]. Para Bédard et al. [19], uma dimensão pode ser convencional ou espacial. A espacial classifica-se em três tipos: não geométrica, geométrica ou mista. A dimensão espacial não geométrica utiliza referências espaciais nominais, ou seja, dados alfanuméricos que representam um local - nomes de cidades, como Campina Grande, Patos e Cabaceiras. Esse tipo de dimensão espacial é o único suportado pelos DWs convencionais. A dimensão espacial geométrica é composta, em todos os níveis hierárquicos e em todos os membros, de formas geométricas (polígonos, linhas, pontos) que são georreferenciadas, o que permite sua visualização em mapas e a realização de consultas espaciais. Na dimensão espacial mista, uma hierarquia pode conter níveis geométricos e não geométricos, ou seja, apenas um subconjunto dos níveis hierárquicos é representado por geometrias georreferenciadas. Os três tipos de dimensões espaciais podem ser vistos na Figura 2, a seguir.

Figura 2 – Os três tipos de dimensões espaciais de acordo com Bédard et al. Extraído de [21]



Já para Han et al. [20], existem três casos ao se modelarem dimensões em cubo de dados espaciais: dimensão não espacial (*nonspatial dimension*), dimensão espacial para não espacial (*spatial-to-nonspatial dimension*) e dimensão espacial para espacial (*spatial-to-spatial dimension*). A dimensão não espacial contém apenas dados não espaciais. A dimensão espacial para não espacial tem dados espaciais apenas nos membros do nível hierárquico mais detalhado, também denominado nível primitivo. As generalizações a partir do nível primitivo não têm dados espaciais. Por fim, na dimensão espacial para espacial, os membros de todos os níveis hierárquicos têm dados espaciais.

2.5.2. Medida espacial

O conceito de medida espacial pode ser introduzido quando o objeto de interesse no processo decisório é a própria informação espacial. A medida espacial pode ser analisada por meio das dimensões espaciais e não espaciais (convencionais). Na literatura, encontram-se diferentes definições para as medidas espaciais.

Han et al. [20] definem medida espacial como um conjunto de referências para um objeto espacial. Bédard et al. [19] propõem três tipos de medida espacial. A primeira consiste de uma geometria ou um conjunto de geometrias obtidas pela combinação de múltiplas dimensões espaciais, ou seja, um conjunto de coordenadas computadas por meio de uma operação espacial como merge, união ou intersecção. A segunda resulta de uma métrica espacial ou de um operador topológico, como área ou distância, por exemplo. A última é um conjunto de apontadores para as geometrias armazenadas em outra estrutura ou tecnologia.

Bimonte et al. [22] referem que uma medida espacial pode ser complexa, ou seja, um objeto espacial como um membro de uma dimensão espacial que contém dados espaciais e convencionais e uma função definida pelo usuário para realizar a agregação dos seus atributos.

As tecnologias GIS, geralmente, definem um objeto geográfico através de um atributo geométrico (polígono, linha, ponto etc.) que pode estar associado a atributos temáticos ou descritivos (nome, classe socioeconômica, etc.). É bastante comum a construção de SDWs em banco de dados relacionais que utilizam a abordagem de objeto geográfico descrita acima.

2.6. LINGUAGEM FORMAL VIZQL

A maioria das ferramentas de análise organiza os dados a serem analisados em tabelas. Isso significa que o usuário deve: 1) selecionar a porção dos dados que deseja analisar; 2) organizar os dados em uma tabela e 3) criar gráficos ou outras representações visuais dos dados a partir da tabela. Uma ferramenta capaz de sumarizar e analisar grandes quantidades de dados em tabelas é a *Pivot Table* [23]. Com ela, o usuário pode configurar ou mudar a estrutura da tabela adicionando campos aos seus eixos (e.g. linhas e colunas). O nome *Pivot Table* deve-se à sua capacidade de permutar os eixos da tabela.

A interface Polaris, proposta por Stolte e Hanrahan [24], é uma extensão da ferramenta *Pivot Table* e dispõe de uma linguagem de especificação visual, denominada VizQL, que descreve as consultas utilizadas para recuperar dados e a forma como são visualizados. Uma das contribuições dessa linguagem é permitir a organização de vários gráficos em tabelas, técnica utilizada por estatísticos na análise de dados [25] [26] [27].

A VizQL é formada por três componentes: a) componente para a especificação da configuração da tabela, b) componente para a especificação do tipo de gráfico que será exibido nas células da tabela e c) componente para a especificação dos detalhes de codificação visual. A Álgebra Tabular (*Table Algebra*), definida em [24], é utilizada para descrever a estrutura da tabela de visualização dos dados, também chamada de “pequenos displays múltiplos” (*small multiple displays*).

O componente para especificar a configuração da tabela dispõe de divisórias denominadas colunas, linhas, filtros e páginas (Figura 3). Duas divisórias – as linhas e as colunas – são utilizadas para particionar os dados em linhas e colunas, formando uma tabela. Cada célula da tabela contém um painel onde os valores dos campos quantitativos serão apresentados. Outra divisória – páginas – é utilizada para particionar os dados em páginas. Cada página contém uma tabela. A divisória “filtros” restringe os dados a serem apresentados na tabela. Assim, somente os valores de medidas que satisfizerem às restrições são exibidos. A ordem dos campos nas divisórias determina a forma como os dados são aninhados, e a configuração de uma divisória especifica uma expressão VizQL.

Na álgebra tabular, os operandos são os nomes originados dos campos qualitativos e quantitativos, para cada um dos quais é atribuído um conjunto ordenado de nomes da seguinte maneira: para um campo qualitativo, os nomes dos membros do campo são atribuídos ao conjunto, e para um campo quantitativo, o nome do campo é atribuído ao conjunto. Por

exemplo, sendo D e E campos qualitativos, M e N , quantitativos, d_i , o nome de determinado membro do campo D , e_i , o nome de certo membro do campo E , M' , o nome do campo quantitativo, e M e N' , do campo quantitativo N , os conjuntos são determinados da seguinte forma:

$$D = \text{domínio}(D) = \{d_1, \dots, d_n\}$$

$$M = \{M'\}$$

Os conjuntos contendo os nomes serão utilizados na configuração da estrutura da tabela, enquanto que os valores dos campos quantitativos (valores das medidas) serão utilizados nos painéis de visualização contidos nas células da tabela.

Os operadores *concatenation* (+), *cross* (x) e *nest* (/) são definidos na álgebra tabular da seguinte forma:

- *Concatenation* (+): realiza a união dos conjuntos ordenados provenientes dos campos. Exemplo:

$$D + E = \{d_1, \dots, d_n\} + \{e_1, \dots, e_n\}$$

$$= \{d_1, \dots, d_n, e_1, \dots, e_n\}$$

$$D + M = \{d_1, \dots, d_n\} + \{M'\}$$

$$= \{d_1, \dots, d_n, M'\}$$

$$M + N = \{M'\} + \{N'\}$$

$$= \{M', N'\}$$

- *Cross* (x): realiza um produto cartesiano nos conjuntos ordenados. Exemplo:

$$D \times E = \{d_1, \dots, d_n\} \times \{e_1, \dots, e_n\}$$

$$= \{d_1e_1, \dots, d_1e_n, \dots, d_2e_1, \dots, d_2e_n, \dots, d_ne_1, \dots, d_ne_n\}$$

$$D \times M = \{d_1, \dots, d_n\} \times \{M'\}$$

$$= \{d_1M', \dots, d_nM'\}$$

- *Nest* (/): similar ao produto cartesiano, mas cria apenas conjuntos para os quais os registros estejam relacionados e pode ser interpretado como “ E dentro de D ”. Por exemplo, *trimestre/mês* retorna o conjunto que contém apenas os meses contidos no trimestre do ano correspondente (e.g. T1Janeiro, T2Abril, etc.). Sendo r um registro da base de dados e $D(r)$ o valor para o campo D do registro r , pode-se definir o operador como:

$$D/E = \{d_i e_j \mid \exists r \in R \text{ st } D(r) = d_i \ \& \ B(r) = b_j\}$$

O conjunto resultante da aplicação dos operadores da álgebra tabular é denominado de conjunto na forma normalizada; cada elemento do conjunto é uma concatenação ordenada de um ou mais valores, e cada divisória é representada por um conjunto na forma normalizada. A intersecção dos elementos, um elemento para cada divisória, identifica um valor de um campo quantitativo.

A expressão VizQL é gerada por meio da adição de campos às divisórias. Ao adicionar campos organizados em diferentes hierarquias, o operador *cross* (X) da álgebra tabular é aplicado aos campos. Caso os campos estejam organizados conforme determinada hierarquia, o operador *nest* (/) é aplicado aos campos.

Na Figura 3, exemplifica-se a utilização da álgebra tabular por meio da ferramenta Tableau (versão comercial do Polaris) onde campos qualitativos (ANO, MÊS DO ANO e DESCRIÇÃO TIPO EMPENHO) e quantitativos (VALOR EMPENHO, QUANTIDADE DE EMPENHOS e MÉDIA VALOR EMPENHO) foram adicionados às divisórias colunas e linhas para a formatação da tabela.

Figura 3 – Aplicação da Álgebra Tabular na ferramenta Tableau

The screenshot shows the Tableau interface with a data table. The columns are ANO, MÊS DO ANO, and DESCRIÇÃO TIPO EMPENHO. The rows are grouped by ANO (2012) and MÊS DO ANO (01 to 05). The table displays values for VALOR EMPENHO, QUANTIDADE EMPENHOS, and MÉDIA VALOR EMPENHO across three categories: ESTIMATIVO, GLOBAL, and ORDINÁRIO.

ANO	MÊS DO ANO	DESCRIÇÃO TIPO EMPENHO	ESTIMATIVO	GLOBAL	ORDINÁRIO
2012	01	VALOR EMPENHO	RS 16089831.54000	RS 17972422.71000	RS 53908228.61000
		QUANTIDADE EMPENHOS	139	391	4.023
		MÉDIA VALOR EMPENHO	RS 130142.67295	RS 45965.27547	RS 13400.90711
	02	VALOR EMPENHO	RS 12207976.57000	RS 21562191.60000	RS 72818616.52000
		QUANTIDADE EMPENHOS	123	371	5.789
		MÉDIA VALOR EMPENHO	RS 99251.84203	RS 58119.11499	RS 12578.44473
	03	VALOR EMPENHO	RS 4242225.51000	RS 16965860.31000	RS 68947291.31000
		QUANTIDADE EMPENHOS	106	384	5.608
		MÉDIA VALOR EMPENHO	RS 40020.99538	RS 44234.01122	RS 12294.45280
	04	VALOR EMPENHO	RS 9698558.76000	RS 6729462.95000	RS 53022919.99000
		QUANTIDADE EMPENHOS	122	157	3.432
		MÉDIA VALOR EMPENHO	RS 79496.38328	RS 42602.82134	RS 15449.56876
	05	VALOR EMPENHO			RS 43905.47000
		QUANTIDADE EMPENHOS			23
		MÉDIA VALOR EMPENHO			RS 1908.93348

2.7. CONSIDERAÇÕES

Neste capítulo, apresentou-se a fundamentação teórica que embasa este trabalho de dissertação. Foram apresentados os principais conceitos presentes nas soluções SOLAP, tornando possível destacar a relevância e a complexidade da análise dos dados espaciais multidimensionais para o processo de tomada de decisão.

A análise dos dados convencionais e espaciais requer toda uma infraestrutura que torna viável a extração de informações úteis para aperfeiçoar o processo de tomada de decisão. Essa infraestrutura geralmente conta com armazéns de dados espaciais (SDWs) capazes de manter uma visão histórica das operações de negócio de uma empresa e de ferramentas SOLAP capazes de sumarizar grandes quantidades de dados em tempo hábil. As ferramentas SOLAP também permitem o uso de operadores convencionais e espaciais para a navegação e a seleção dos dados e possuem interfaces intuitivas e interativas voltadas para os analistas, facilitando o processo de análise.

No capítulo seguinte, será apresentado o estado da arte em análise SOLAP.

3. TRABALHOS RELACIONADOS

Neste capítulo, são apresentados alguns trabalhos relacionados ao tema *Spatial OLAP* encontrados na literatura.

3.1. JMAP SPATIAL OLAP

Rivest et al. [4] revisam os conceitos que fundamentam os sistemas SOLAP e apresentam uma ferramenta denominada *JMap Spatial OLAP*, cuja principal característica é o suporte para a geovisualização, a interatividade e a animação.

A ferramenta é baseada na arquitetura ROLAP e apresenta três tipos de dimensões espaciais: a) não geométrica, b) geométrica e c) mista. As medidas espaciais são provenientes da união ou intersecção de geometrias, formando um conjunto de geometrias. A ferramenta não dá suporte a medidas espaciais provenientes de operadores métricos e topológicos (ex. distância, área etc.). Os dados espaciais não são armazenados de forma integrada com o banco de dados multidimensional, pois ponteiros fazem referência aos dados espaciais armazenados em certo formato/tecnologia.

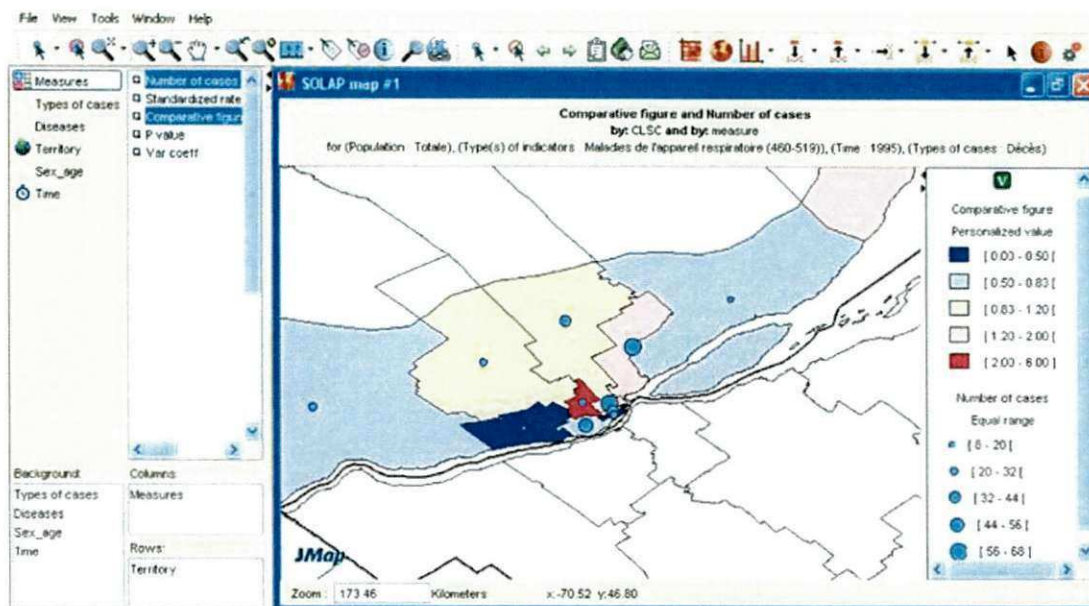
A exploração dos dados, que podem ser visualizados por meio de tabelas, gráficos e mapas sincronizados, é feita por meio de *displays* interativos voltados para o usuário final, e os mapas podem ser temáticos ou com gráficos sobrepostos. Quanto à visualização, classificam-se como simples ou múltiplos (exibidos em um mesmo display e usados para facilitar a comparação).

Legendas interativas podem ser criadas e exibidas e estão relacionadas com três componentes: tempo, espaço e medida. A ferramenta também dispõe de uma linha de tempo denominada *drillable timeline*, que auxilia o usuário a controlar a visualização das dimensões temporais e suportar operações *drill-down* e *roll-up* diretamente do cursor de tempo. Essa linha de tempo permite a visualização de mapas animados, que facilitam a visualização da evolução do fenômeno de interesse.

Os seguintes operadores SOLAP foram implementados para a exploração dos dados espaciais: *spatial drill-down*, *spatial roll-up*, *spatial drill-across*, *open* e *close*. Os operadores *open* e *close* são variantes do *drill-down* e *roll-up*, respectivamente, mas diferem pelo fato de que mantêm o contexto dos outros membros das dimensões. Uma desvantagem dessa ferramenta é que ela não admite o uso de operadores espaciais (métricos ou topológicos).

A Figura 4 mostra um exemplo da interface do *JMAP Spatial OLAP* onde um estudo sobre doenças respiratórias na província de Quebec, Canadá, é realizado.

Figura 4 – Interface da ferramenta *JMAP Spatial OLAP*



3.2. GeoWOLAP

Bimonte et al. [5] apresentam a ferramenta GeoWolap, cuja principal característica é o suporte a medidas espaciais complexas, ou seja, uma dimensão espacial com objetos espaciais e dados alfanuméricos pode se tornar uma medida. Com base no modelo GeoCube [22], o GeoWolap permite navegação interativa em um cubo.

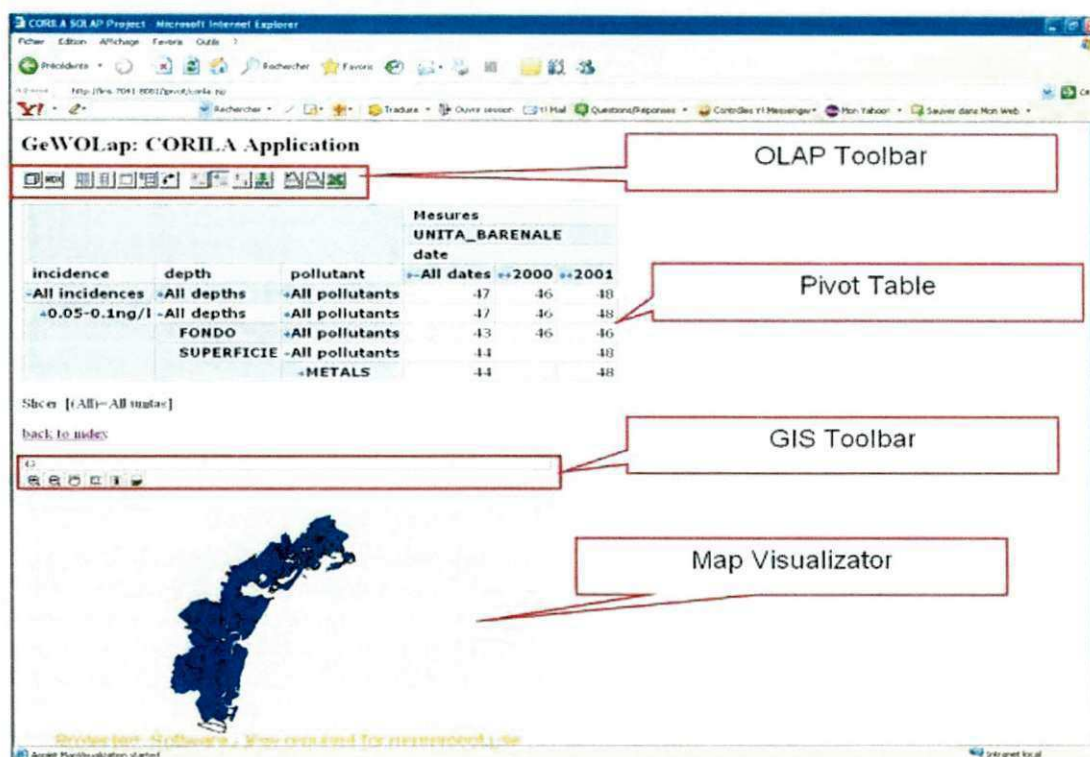
Sua arquitetura é composta de três camadas: camada de dados, servidor SOLAP e camada cliente (interface com o usuário). A camada de dados foi implementada com o uso do banco de dados objeto-relacional Oracle (ORDBMS), onde os dados convencionais e espaciais são armazenados. O GeoWolap utiliza o Mondrian [28] como servidor SOLAP, e suas funcionalidades foram estendidas para suportar medidas complexas por meio do uso de funções de agregação definidas em PL/SQL no ORACLE, e as funções de agregação para medidas complexas foram delegadas ao ORDBMS.

O GeoWOLAP dispõe de interface Web e foi implementado com JPvito e MapExtrema Java, permitindo a visualização sincronizada dos resultados na forma tabular e na cartográfica. A interface é composta de quatro painéis principais. O primeiro contém uma barra de ferramenta OLAP para a seleção dos metadados do cubo (dimensões e medidas), a

seleção de operadores OLAP (*drill-down*, *roll-up*, etc.), entre outras opções; o segundo contém uma tabela de pivotemaneto (*Pivot Table*); o terceiro, uma barra de ferramenta GIS (*zoom in*, *zoom out*, *pan*, *info*, etc.), e o quarto, um mapa (Figura 5).

A ferramenta proposta pelos autores não permite o uso de operadores espaciais (métricos ou topológicos) e utiliza tecnologias proprietárias.

Figura 5 – Interface da ferramenta GeWOlap



3.3. SOVAT

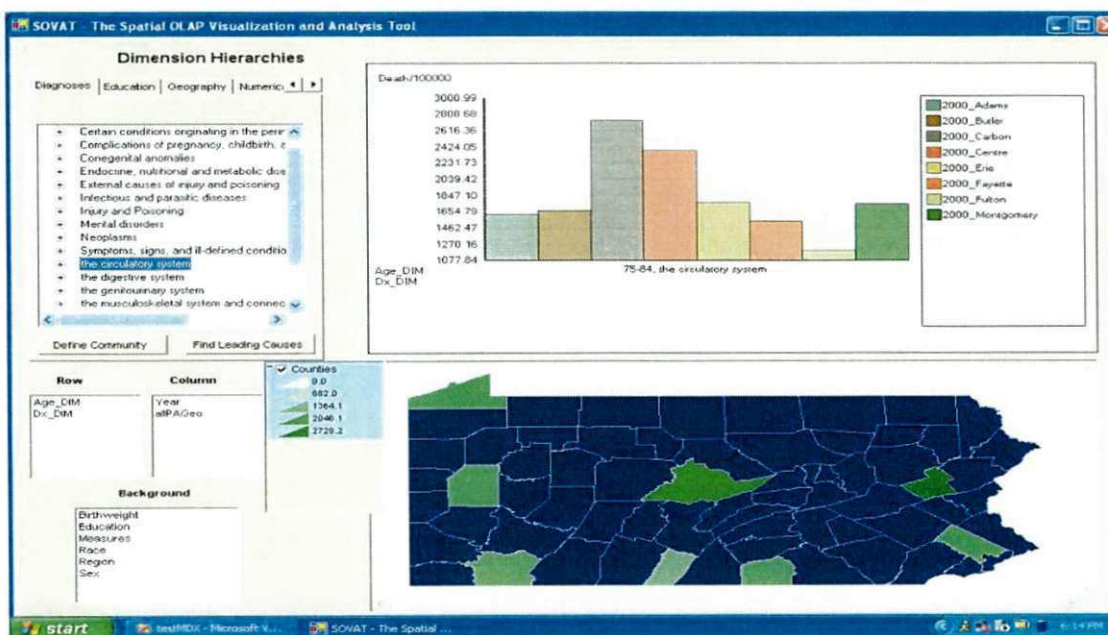
Em [6], Scotch e Parmanto propõem a ferramenta SOVAT (*Spatial OLAP Visualization and Analysis Tool*) para auxiliar pesquisadores e profissionais da área de saúde pública no processo de tomada de decisão. A ferramenta é composta de três camadas: a camada de dados, a camada de aplicação e a camada de cliente. A primeira implementa a abordagem federada para o *Data Warehouse*, onde os dados multidimensionais e espaciais são mantidos em bases distintas; já a segunda é responsável por integrar os dados espaciais e multidimensionais, além de dispor de mecanismos para permitir a análise dos dados integrados; a camada de cliente consiste de uma aplicação *desktop* implementada na plataforma .NET, que conduz o usuário a definir consultas apenas arrastando e soltando (*drag*

and drop) componentes e apresenta os resultados das consultas em gráficos e mapas (Figura 6). Com a aplicação não é possível comparar os dados por meio de displays múltiplos e, por ser desktop, não permite que usuários acessem remotamente pela Internet.

Dentre as características apresentadas como as mais importantes da ferramenta, estão: habilidade para lidar com grandes quantidades de dados, realizar análises estatísticas, permitir a exploração detalhada dos dados, visualizar os dados por meio de gráficos e mapas e realizar análise espacial.

A ferramenta suporta operações como *roll-up*, *drill-down*, *slice and dice* e *pivot*. Outra importante característica citada foi a apresentação de um novo operador denominado de *drill-out*, que busca resolver consultas envolvendo detecção de fronteiras de geometria. Também é possível utilizar o operador espacial *Buffer* para transformar as geometrias e calcular a melhor rota. Porém, a ferramenta não dispõe de operadores espaciais métricos ou topológicos.

Figura 6 – Interface da ferramenta SOVAT



3.4. PIET

Escribano et al. [7] propõem uma ferramenta denominada Piet, que integra tecnologias GIS e OLAP e é capaz de realizar a pré-computação das camadas (*overlays*) do mapa. Essa ferramenta utiliza a abordagem federada para integrar os dados multidimensionais e espaciais

onde um DW armazena os dados convencionais, e um banco de dados geográfico armazena os dados espaciais.

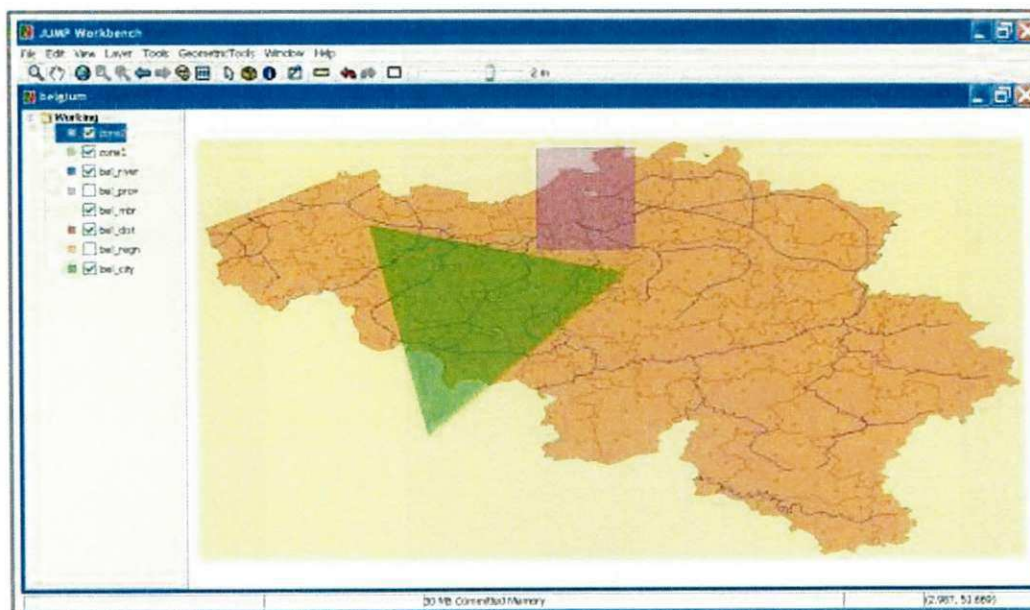
A arquitetura do Piet é composta de três camadas: camada de dados, servidor SOLAP e camada cliente. O servidor é formado por dois módulos: o gerador de dados pré-calculados e o processador de consultas.

O gerador de dados pré-calculados contém as subgeometrias resultantes da fase denominada *subpolygonization*, os valores de fatos associados a essas subgeometrias, as camadas pré-computadas, os metadados que descrevem a estrutura do DW, dos mapas e suas associações e as relações GIS-OLAP que contém as informações necessárias para associar os componentes geométricos às informações armazenadas no DW. A fase de *subpolygonization* decompõe cada camada temática em polígonos convexos abertos.

O processador de consultas permite o processamento de quatro tipos de consulta: geométrica, agregação geométrica, OLAP e GIS-OLAP. Uma linguagem denominada GISOLAP-QL é proposta. Essa linguagem de consulta é composta de duas partes: a primeira parte para recuperar os dados espaciais, e a segunda, para recuperar os dados multidimensionais.

A ferramenta Piet não tem uma interface interativa que omita detalhes da linguagem de consulta (Figura 7). Piet é composto por duas aplicações, uma aplicação Web para consultas OLAP e outra *desktop* para consultas espaciais, logo, não tem uma visualização integrada dos dados multidimensionais e espaciais.

Figura 7 – Interface desktop da ferramenta PIET



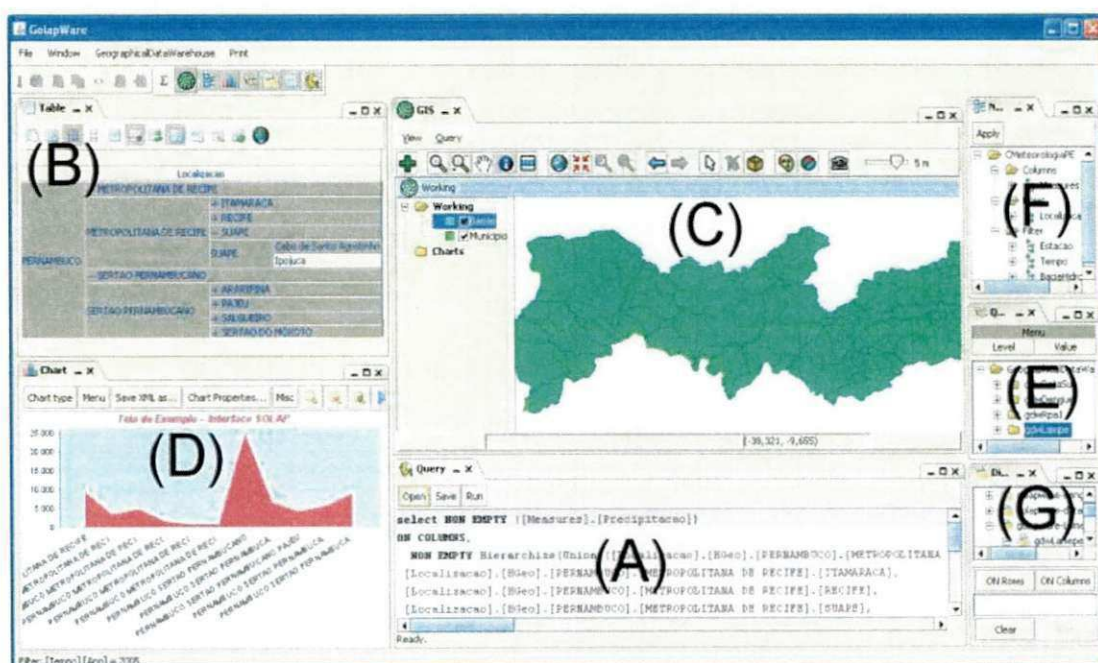
3.5. GOLAPWARE

Em [8], foi desenvolvida uma instância da arquitetura GOLAPA [29], denominada de *Golapware*, utilizada para o processamento da *GeoMDQL*, que é uma linguagem de consulta geográfico-multidimensional para análise SOLAP. Essa linguagem é uma extensão da linguagem multidimensional MDX e provê operadores espaciais e multidimensionais de forma integrada.

A arquitetura GOLAPA dispõe de três camadas essenciais para o processamento SOLAP: camada de dados (*Geographical Data Warehouse - GDW*), servidor SOLAP e interface com o usuário. O GDW foi implementado no SGBD PostgreSQL, com a extensão espacial PostGis. A ferramenta GeoDWCASE [30] foi utilizada para modelar, validar e gerar esquemas para GDW. O servidor SOLAP é uma extensão do OLAP Mondrian. O servidor contém o engenho responsável pelo processamento SOLAP, denominado GOLAPE (*Geographical Online Analytical Processing Engine*), que serve para dar suporte às consultas espaciais feitas por meio da linguagem GeoMDQL.

A interface com o usuário promove a visualização integrada dos dados multidimensionais e espaciais e foi implementada utilizando duas abordagens: *Web* e *Desktop*. Na interface, o usuário deve especificar consultas na linguagem GeoMDQL que permite a utilização dos operadores espaciais propostos pelo OGC.

Figura 8 – Interface desktop da ferramenta Golapware



A ferramenta Golapware não oferece componentes de interface para que o usuário possa interagir e definir as consultas de forma visual, o que dificulta a análise dos dados.

A interface desktop da ferramenta *Golapware* é exibida na Figura 8. Ela é composta de sete componentes: (A) painel para a especificação textual das consultas *GeoMDQL*, (B) painel que exhibe as tabelas multidimensionais resultantes das consultas, (C) painel que exhibe os mapas resultantes das consultas, (D) painel para a construção e exibição de gráficos, (E) painel que exhibe as consultas salvas anteriormente, (F) painel que permite explorar as dimensões, fatos, hierarquias, níveis e membros do cubo e (G) painel que exhibe os cubos disponíveis em outros catálogos registrados na interface.

3.6. MAPWAREHOUSE

Cardoso [3] propõe um framework para que gestores possam definir cubos de *Data Warehouse* Espacial e realizar consultas sobre eles. O framework é denominado Mapwarehouse e dispõe de linguagens gráficas para definição de esquemas e de consultas.

A arquitetura do framework é composta de três camadas: Cliente, Aplicação e *Data Warehouse*. A camada Cliente foi desenvolvida utilizando-se as tecnologias Java Applet e Java Server Pages e compreende o conjunto de interfaces gráficas necessárias para: a) definição de esquemas conceituais de DW, b) definição de consultas conceituais e c) visualização dos resultados.

Para a criação de esquemas conceituais de DW, foi proposto um metamodelo que integra totalmente os dados espaciais e convencionais ao DW e que considera o DW Espacial uma extensão natural de um DW convencional, o que significa que os conceitos tradicionais como dimensões, hierarquias, fatos e medidas foram estendidos para lidar com dados espaciais que podem ser utilizados tanto em dimensões quanto em medidas (Figura 9).

Outro metamodelo (Figura 10) foi proposto para definir as consultas conceituais que, com o auxílio de interfaces gráficas, abstraem a necessidade de os usuários dominarem uma linguagem de consulta lógica (e.g. SQL). A linguagem de consulta permite a construção de restrições utilizando-se os operadores espaciais, o que aumenta a abrangência das consultas.

Os resultados podem ser visualizados em tabelas e mapas de forma sincronizada. O mapa permite a interação com o usuário por meio das funções *zoom in*, *zoom out* e *pan* e, também, é possível navegar nas hierarquias. Um exemplo da interface Web da ferramenta é exibido na Figura 11.

Figura 9 – Metamodelo do Mapwarehouse. Extraído de [3]

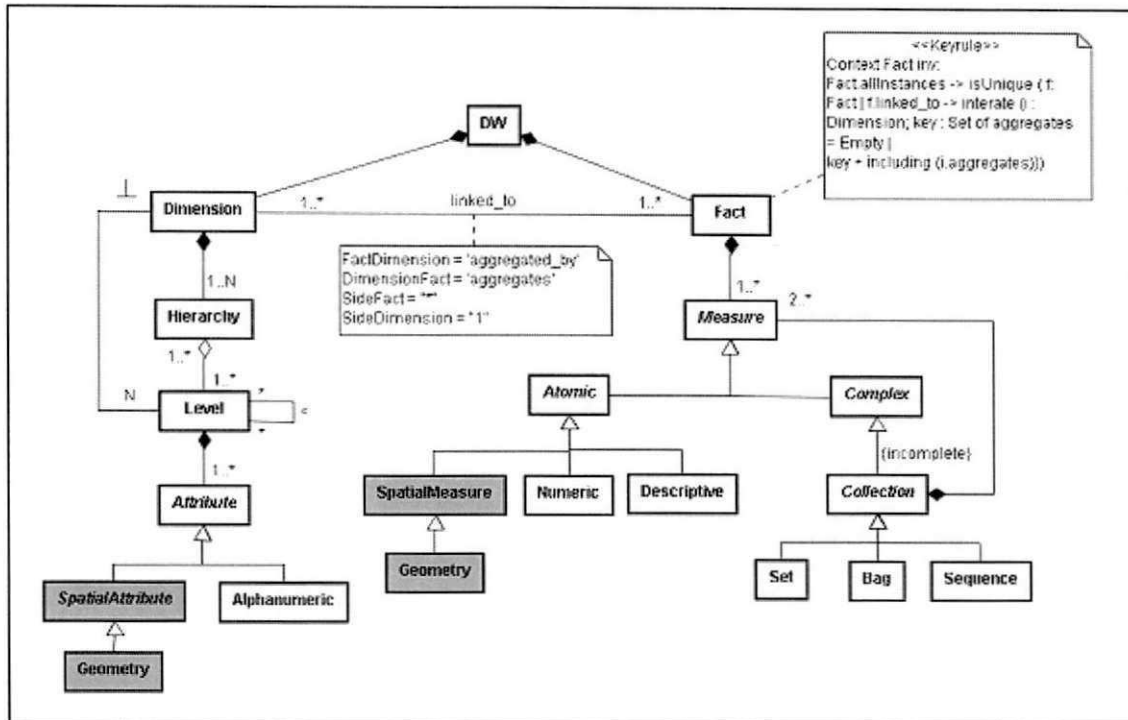
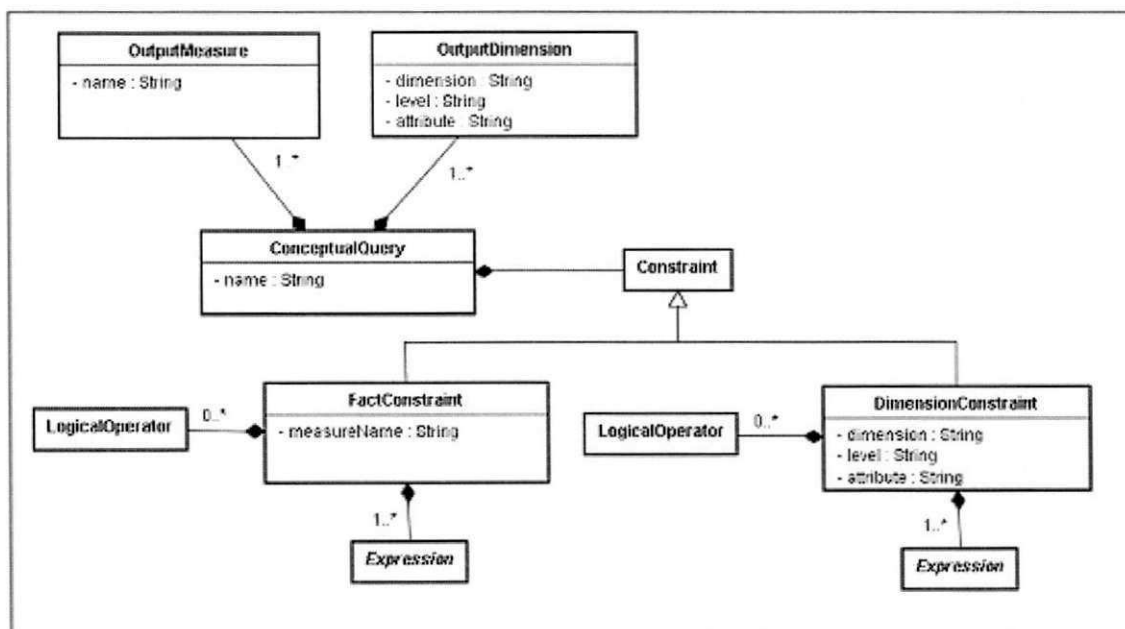


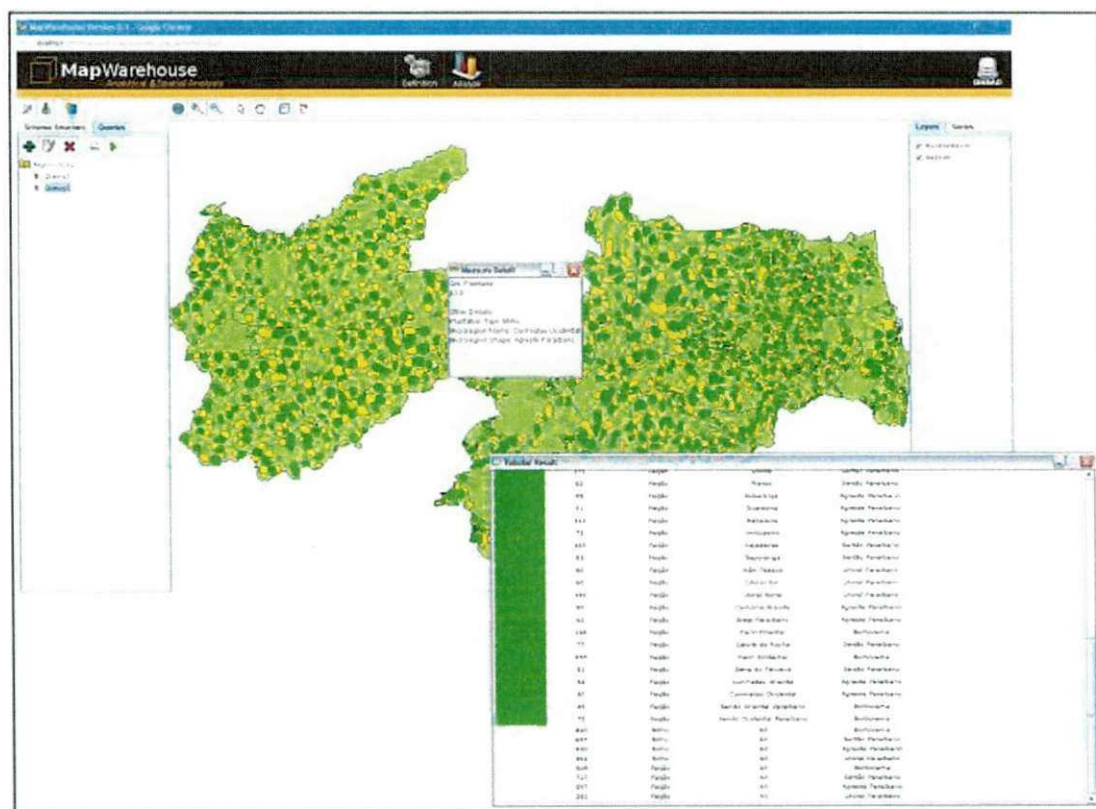
Figura 10 - Diagrama de classes para a Consulta Conceitual. Extraído de [3]



Na camada de aplicação, está inserido o engenho principal responsável pelo funcionamento da ferramenta e que serve de interface entre a camada cliente e o *Data Warehouse*. O engenho foi implementado na tecnologia Java e é capaz de se comunicar com DWs armazenados em SGBDs de diversos fabricantes. Porém, para que ocorra a comunicação, deve-se estender a solução com as particularidades do SGBD do fabricante. Por fim, na camada de *Data Warehouse*, encontram-se os dados convencionais e espaciais que são armazenados no mesmo ambiente, numa abordagem integrada, o que exige que o SGBD escolhido tenha a capacidade de armazenar e manipular ambos os tipos de dados.

O cliente da solução proposta por Cardoso é fortemente acoplado à camada de Aplicação, o que não permite a análise de cubos pertencentes a outros servidores SOLAP.

Figura 11 – Interface da ferramenta Mapwarehouse



3.7. POLARIS

Stolte e Hanrahan [24] apresentam uma interface para a análise de dados que estende a interface *Pivot Table*. Polaris é uma interface empregada na análise exploratória de grandes bases de dados relacionais multidimensionais, que tem um conjunto de componentes gráficos

para especificar visualmente as consultas relacionais e a visualização dos dados (linguagem de especificação visual).

A especificação visual consiste de três componentes: a especificação da configuração da tabela, do tipo de gráfico apresentado em cada célula da tabela e dos detalhes de codificação visual. Para a configuração das tabelas, a ferramenta dispõe de três componentes gráficos denominados prateleiras (colunas, linhas e camadas). Cada prateleira consiste de uma expressão em um formalismo algébrico denominado de álgebra tabular [24].

O sistema Polaris ajuda os analistas a construir gráficos de forma flexível, especificando os componentes individuais do gráfico. Ao especificar a configuração da tabela, o usuário, implicitamente, especifica os eixos associados aos gráficos que podem ser classificados em três famílias: ordinal-ordinal, ordinal-quantitativo e quantitativo-quantitativo.

Dois componentes estão disponíveis para a codificação visual: o primeiro determina o tipo de marcação (retângulo, círculo, texto, linha, x-barra, y-barra etc.), e o segundo codifica o campo a uma propriedade de retina (forma, tamanho, cor etc.). A interface da ferramenta Polaris é ilustrada na Figura 12.

A grande vantagem da interface Polaris é permitir a análise visual por meio da linguagem de especificação visual denominada de VizQL. Porém, essa linguagem não permite o uso de operadores espaciais para a construção de filtros e dá suporte apenas a campos espaciais do tipo ponto (latitude e longitude). Embora dê suporte à sobreposição de camadas de contexto, onde as camadas são originadas de outras fontes de dados, não permite a sobreposição de camadas originadas dos campos geométricos. A solução proposta nesta dissertação estende a linguagem VizQL para a análise visual. Essa extensão visa suprir as desvantagens encontradas nessa linguagem para a análise espacial.

A ferramenta foi construída para bases de dados relacionais, por isso não dá suporte a hierarquias e níveis, conceitos relacionados aos cubos multidimensionais, nem permite o uso de operadores espaciais.

O sistema Polaris evoluiu para um sistema proprietário que, atualmente, é comercializado como uma ferramenta denominada Tableau (Figura 13).

Figura 12 – Interface da ferramenta Polaris

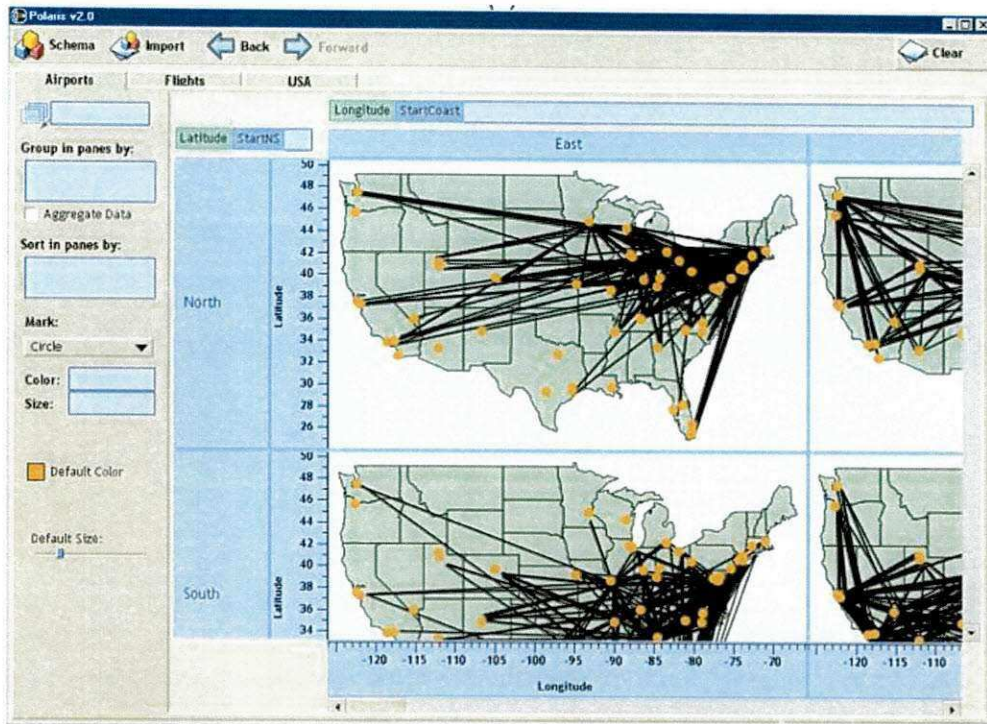
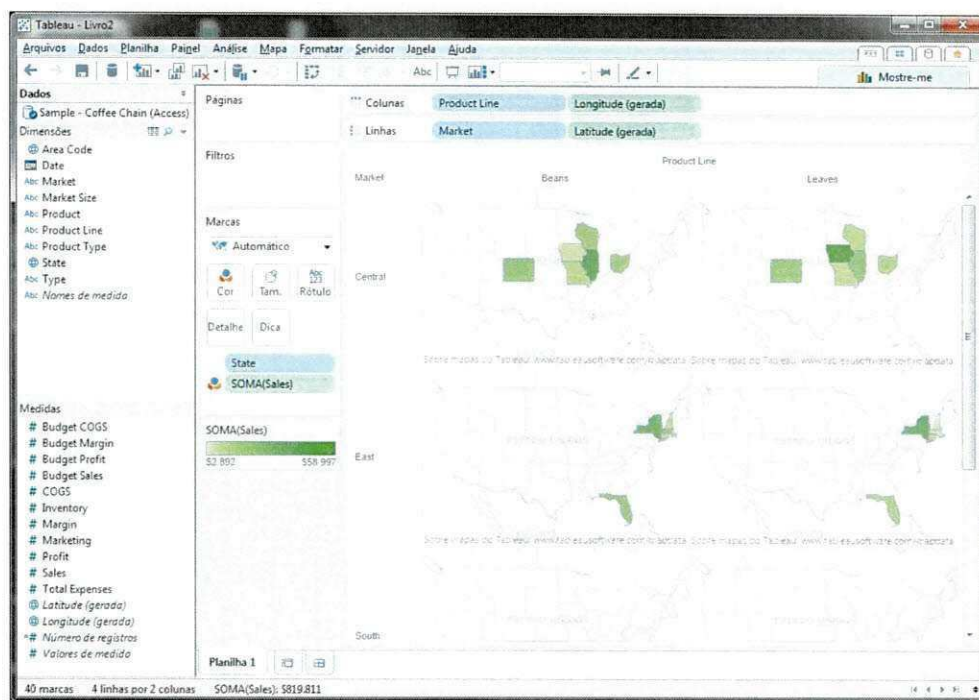


Figura 13 – Interface da ferramenta Tableau



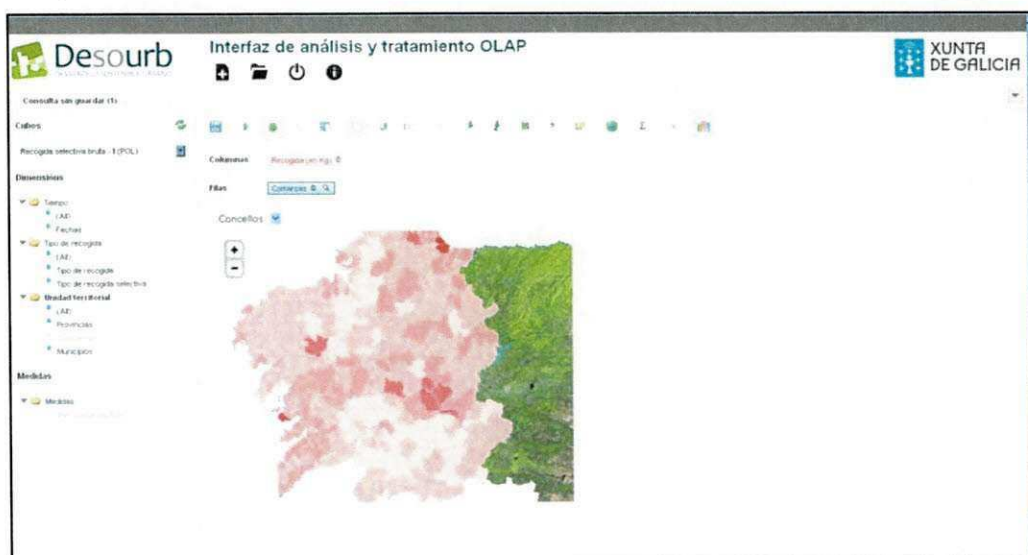
3.8. DESOURB/Saiku

Lamas et al. [31], juntos com o Instituto de Estudos do Território da junta de Galícia, no âmbito do projeto DESOURB, desenvolveram uma integração do servidor de mapas *MapServer* com a ferramenta de análise (OLAP) *Saiku Analytics*, possibilitando a visualização dos dados com dimensão espacial em mapas.

Com a ferramenta Web, podem-se criar mapas temáticos que representam a distribuição espacial de determinada medida do cubo por meio de diferentes tons de cor. A graduação da cor representa diferentes intervalos da medida selecionada em unidades territoriais. Já a ferramenta Saiku oferece uma interface para a navegação e a seleção dos metadados do cubo (medida, dimensão, etc.) que são utilizados para se construir o mapa temático. Para esse propósito, essa ferramenta disponibiliza dois componentes (colunas e linhas), com os quais o usuário pode selecionar a dimensão e a medida que deseja analisar, respectivamente. Não é possível criar restrições espaciais para a construção de mapas, pois a ferramenta não disponibiliza operadores espaciais.

As tecnologias utilizadas foram: servidor SOLAP Geomondrian, ferramenta OLAP Saiku, cliente de mapa LeafLet, servidor de mapa MapServer, biblioteca para a construção de legendas ColorBrewer e banco de dados PostgreSQL com a extensão espacial PostGIS. Com essa ferramenta, não se podem analisar cubos provenientes de outras fontes multidimensionais, apenas os provenientes do servidor GeoMondrian. Na Figura 14, exibe-se sua interface.

Figura 14 – Interface da extensão da ferramenta SAIKU para o projeto Desourb



3.9. CONSIDERAÇÕES

A integração dos dados espaciais com as ferramentas OLAP traz grandes benefícios para o processo de tomada de decisão, permitindo a exploração desses dados para a descoberta de novas informações, o que, antes, não possível. Com base nas características fundamentais observadas nas soluções investigadas, conclui-se que as ferramentas SOLAP devem:

- I. Permitir a criação de consultas por meio de uma linguagem de especificação visual, com a qual o usuário pode conduzir a análise de forma intuitiva, sem ter que dominar uma linguagem de consulta específica. Essa característica foi observada nos trabalhos [3], [4], [5], [6], [24] e [31];
- II. Ter uma visualização integrada dos dados multidimensionais e espaciais. O uso do mesmo ambiente para visualizar os dados multidimensionais e espaciais diminui o esforço realizado para a análise dos dados. Essa característica foi observada nos trabalhos [3], [4], [5], [6], [8], [24] e [31];
- III. Ter uma interface Web para que o usuário possa acessar remotamente a solução por meio de um navegador. Essa característica foi observada nos trabalhos [3], [4], [5], [8], [24] e [31];
- IV. Permitir a comparação dos dados. O uso de *displays* múltiplos ou de múltiplos mapas facilita a análise ao permitir a visualização dos dados em diferentes perspectivas como, por exemplo, acompanhar a evolução de determinado fenômeno de estudo ao longo dos anos. Essa característica foi observada nos trabalhos [4], [5], [24];
- V. Dar acesso a várias fontes de dados multidimensionais (servidores de cubos). A análise de várias fontes de dados em um mesmo ambiente diminui custos em TI e aumenta a possibilidade de se extraírem informações úteis para melhorar o processo de tomada de decisão das corporações. Essa característica é proposta nesta dissertação;
- VI. Possibilitar a geocodificação dos dados, permitindo a análise espacial em fontes puramente OLAP. Caso os dados espaciais e multidimensionais não estejam em um mesmo ambiente, a geocodificação contribui para que dados espaciais sejam

relacionados aos dados multidimensionais e aumente o poder de análise dos dados. Essa característica é proposta nessa dissertação;

- VII. Dar suporte aos operadores espaciais. Os operadores espaciais possibilitam análises mais abrangentes, uma vez que exploram os componentes espaciais dos dados para a extração de informações. Essa característica foi observada nos trabalhos [3] e [8];
- VIII. Utilizar apenas tecnologias abertas para diminuir custos relacionados à implantação e extensão das funcionalidades da solução. Essa característica foi observada nos trabalhos [3], [4] e [8];
- IX. Ser extensível, para que novas funcionalidades possam ser adicionadas. Por exemplo, a adição de novos operadores, de novas formas de visualização dos dados ou para permitir o acesso a fontes de dados de diferentes tecnologias. Essa característica foi observada nos trabalhos [3] e [4];
- X. Permitir a visualização dos dados por meio de mapas, tabelas e gráficos. A visualização dos dados em várias perspectivas aumenta o poder de análise das soluções SOLAP. Essa característica foi observada nos trabalhos [4], [6], [8], [24] e [31].

As características supracitadas serão utilizadas para comparar os trabalhos abordados. Para isso, um quadro comparativo é exibido na Tabela 1, onde uma célula contendo um X significa que a ferramenta apresenta a característica correspondente e a contendo um – (hífen) significa que a ferramenta não apresenta a característica correspondente.

Tabela 1 – Quadro comparativo entre as soluções investigadas

Soluções	Características									
	I	II	III	IV	V	VI	VII	VIII	IX	X
JMAP Spatial OLAP	X	X	X	X	-	-	-	X	X	X
GeoWOLAP	X	X	X	X	-	-	-	-	-	-
SOVAT	X	X	-	-	-	-	-	-	-	X
Piet	-	-	-	-	-	-	-	-	-	-
Golapware	-	X	X	-	-	-	X	X	-	X
Mapwarehouse	X	X	X	-	-	-	X	X	X	-
Polaris	X	X	X	X	-	-	-	-	-	X
DESOURB/Saiku	X	X	X	-	-	-	-	X	-	X

Todos os trabalhos abordados neste capítulo têm em comum uma arquitetura em que o cliente está fortemente acoplado ao servidor SOLAP, o que impossibilita a análise de dados provenientes de outros servidores SOLAP.

As soluções Piet e Golapware não apresentam interfaces interativas por meio das quais o usuário possa especificar sua consulta. Isso o obriga a dominar uma linguagem de consulta específica. A solução Piet tem duas interfaces distintas: uma para a análise multidimensional e outra para a análise espacial, ou seja, não permite uma visão integrada dos dados. As soluções SOVAT e Piet não dispõem de uma interface Web, por isso não é possível o acesso remoto via Internet. As soluções GeoWOLAP, Piet e Mapwarehouse não dispõem de uma ou mais formas de visualização dos dados: tabelas, gráficos e mapas.

Com exceção das soluções Golapware e Mapwarehouse, as demais ferramentas não permitem o uso de operadores espaciais para a realização de *Slice and Dice* nos cubos de dados, o que diminui a abrangência e o poder das consultas. As soluções abordadas dão suporte apenas a um tipo de integração dos dados, integrada ou federada. Com exceção das soluções Piet e Golapware, as demais têm uma linguagem para a especificação visual da consulta. Porém, apenas a solução Polaris define uma linguagem de especificação para a definição de consultas e a visualização dos dados. As soluções GeoWOLAP, SOVAT, Piet e Polaris não utilizam tecnologias abertas, e a GeoWOLAP, a SOVAT, a Piet e a Golapware não são extensíveis.

Muito esforço é despendido na construção de ferramentas SOLAP, porém, nas soluções investigadas, não é possível reutilizar a camada cliente por falta da padronização da comunicação entre cliente e servidor SOLAP. Assim, com base nas características desejáveis, é notório que nenhuma solução abordada nesta dissertação pode ser considerada satisfatória.

4. UMA PROPOSTA DE *FRAMEWORK* PARA A ANÁLISE ESPACIAL DE FONTES DE DADOS MULTIDIMENSIONAIS

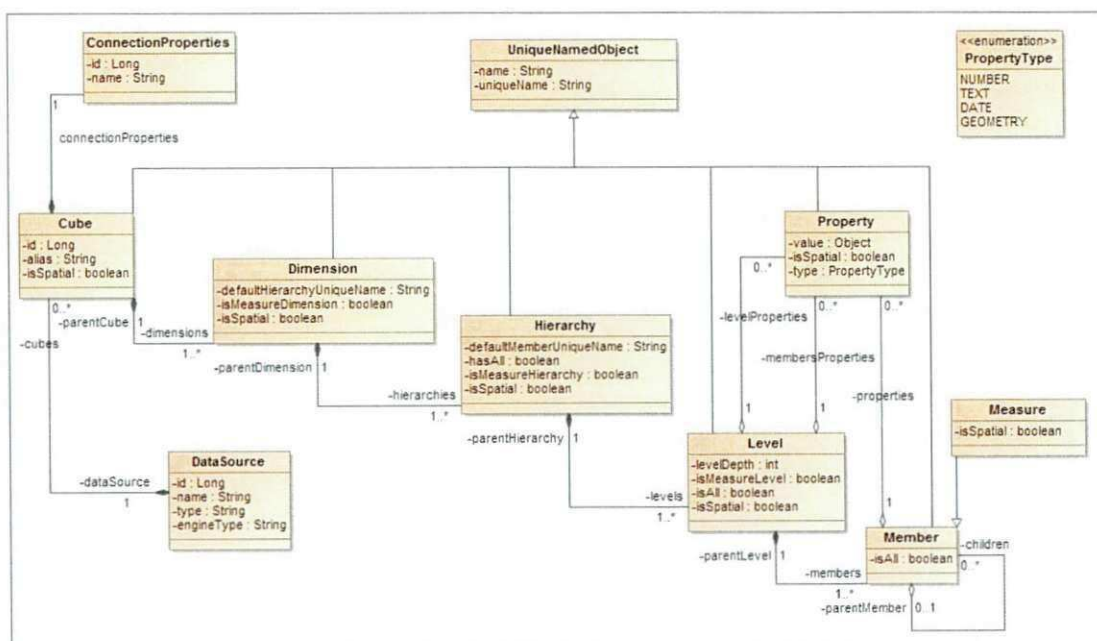
Este capítulo apresenta uma proposta de *framework* para clientes SOLAP, que possibilita a análise dos dados provenientes de diferentes bases de dados espaciais multidimensionais de forma individual, armazenadas em diferentes tecnologias e que atendem aos requisitos apresentados na seção 3.9.

Para alcançar os requisitos sugeridos para o *framework*, definiu-se um modelo para cubos espaciais e se propôs uma extensão da linguagem de especificação visual VizQL [24]. O modelo para cubos espaciais e a extensão da VizQL são apresentados a seguir. Logo depois, detalham-se a arquitetura do *framework* e os pontos de extensão.

4.1. MODELO PARA CUBOS ESPACIAIS

Para o *framework*, um cubo é considerado uma estrutura de dados composta por medidas e dimensões. As dimensões são responsáveis por categorizar os dados do cubo e têm hierarquias que são compostas por níveis. Um nível, por sua vez, é composto por membros. O diagrama de classes que ilustra o cubo pode ser observado na Figura 15.

Figura 15 – Diagrama de classes para o cubo espacial.



Em cada dimensão do cubo, os membros são agrupados em níveis, que são organizados em hierarquias. A hierarquia define diferentes níveis de detalhes. Em uma hierarquia, os níveis respeitam uma ordem, que é representada pela profundidade do nível na hierarquia. Os membros se relacionam, ou seja, um membro de determinado nível é filho de outro pertencente a um nível superior.

Níveis e membros têm propriedades, as quais têm um tipo e um valor. O tipo da propriedade caracteriza o domínio da propriedade e pode ser número, texto, data, ou geometria. O valor é um elemento do domínio da propriedade. O nível lista as propriedades comuns aos membros, assim, é possível saber com antecedência quais são as propriedades dos membros.

As medidas são os valores quantitativos utilizados para mensurar características do fenômeno analisado. No cubo, elas são organizadas em uma dimensão especial, denominada de dimensão de medidas (*measures dimension*), cujos membros são os nomes das medidas e são organizados em uma hierarquia especial, chamada de hierarquia de medidas (*measures hierarchy*). Essa hierarquia tem um nível especial denominado de nível de medidas (*measures level*). No modelo proposto, os atributos *isMeasureDimension*, *isMeasureHierarchy* e *isMeasureLevel*, das classe *Dimension*, *Hierarchy* e *Level* respectivamente, são responsáveis por identificar a dimensão, a hierarquia e o nível de medidas.

Cada um dos componentes do cubo deve ter, pelo menos, dois atributos: nome único e nome. O nome único é utilizado para identificar, de forma unívoca, o componente do cubo, e o nome, para a apresentação ao usuário. Para cada membro da dimensão de medida está associado um conjunto de valores de medida. Existe um valor de medida para cada combinação possível de diferentes dimensões. No *framework* proposto, os valores das medidas não estão associados ao cubo, mas ao resultado de uma consulta multidimensional (Figura 44).

Se os membros de um nível tiverem uma propriedade espacial, eles são considerados espaciais; se a hierarquia tiver um nível espacial, é considerada espacial, e se uma dimensão tiver uma hierarquia espacial, é considerada espacial. Por fim, um cubo é espacial se tiver, pelo menos, uma medida espacial ou uma dimensão espacial.

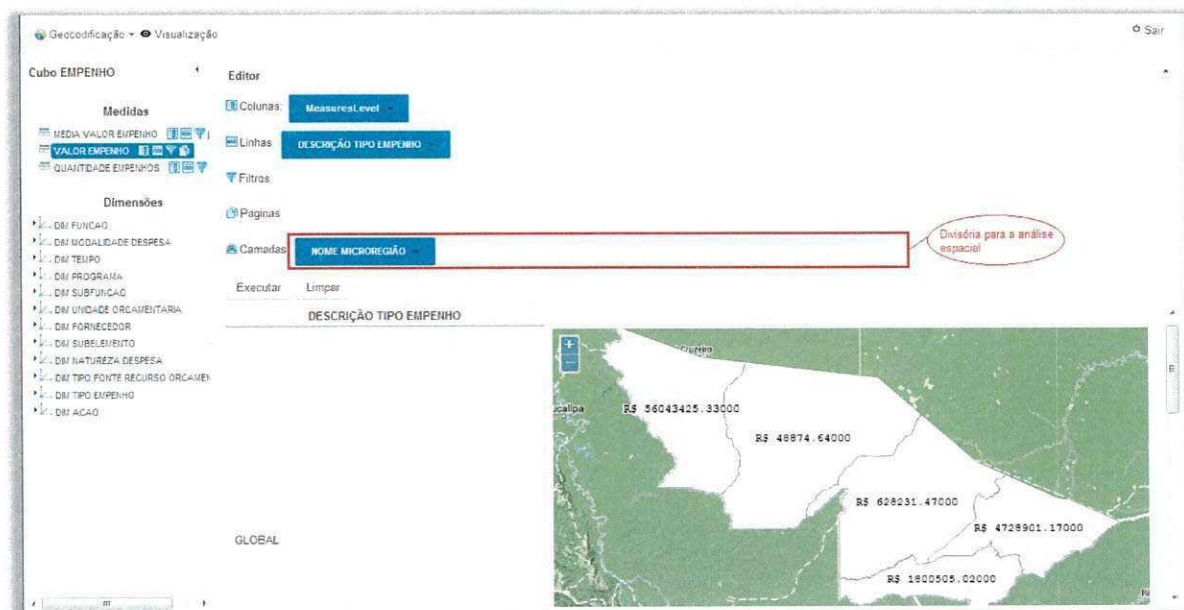
4.2. EXTENSÃO DA LINGUAGEM VIZQL

A interface da solução proposta nesta dissertação é baseada na interface denominada Polaris, proposta por Stolte e Hanrahan [24]. Com o objetivo de facilitar o processo de análise dos dados, a interface do *framework* proposto aqui dispõe de uma linguagem de especificação visual que é uma extensão do formalismo VizQL [24], na qual foram adicionadas especificações para a recuperação e a visualização dos dados espaciais em bases multidimensionais.

A álgebra tabular apresentada na seção 2.6 é utilizada pela linguagem VizQL para particionar os dados de acordo com a especificação visual definida para a configuração da tabela, ou seja, as expressões VizQL das divisórias. As divisórias disponíveis na VizQL são: Colunas, Linhas, Filtros e Páginas.

Para dar suporte à análise espacial, foi adicionada à linguagem VizQL a divisória “Camadas” (Figura 16). No Polaris há também uma divisória denominada Camadas, porém ela é apenas para segmentar os dados e passou a ser denominada Páginas na versão comercial do Polaris (Tableau).

Figura 16 – Divisória Camadas para a análise espacial



Um conceito fundamental em Sistemas de Informação Geográfica (SIG) é a sobreposição de camadas, denominada de *overlay*. Em um SIG, os dados são organizados em

camadas, que podem ser sobrepostas, o que facilita a descoberta de novas informações, pois os dados podem ser mais facilmente correlacionados. Em um cubo, os dados são organizados em níveis hierárquicos. Assim, um nível espacial pode ser utilizado como a camada de um mapa. Porém, não há uma divisória na linguagem VizQL que organize os dados de forma a possibilitar a sobreposição de camadas.

Também foi adicionado um novo operador à álgebra tabular, denominado de *SpatialConcatenation*. Para esse novo operador, os operandos devem ser campos espaciais e os nomes dos campos são atribuídos ao conjunto de nomes resultante da aplicação desse operador, independentemente do tipo do campo (quantitativo ou qualitativo).

Assim, sendo $D_{espacial}$ e $E_{espacial}$ campos qualitativos espaciais, $M_{espacial}$ e $N_{espacial}$ campos quantitativos espaciais, $D'_{espacial}$ o nome do campo qualitativo espacial $D_{espacial}$, $E'_{espacial}$ o nome do campo qualitativo espacial $E_{espacial}$, $M'_{espacial}$ o nome do campo quantitativo espacial $M_{espacial}$, e $N'_{espacial}$ o nome do campo quantitativo espacial $N_{espacial}$. Os conjuntos são determinados da seguinte forma:

$$\begin{aligned} D_{espacial} &= \{D'_{espacial}\} \\ M_{espacial} &= \{M'_{espacial}\} \end{aligned}$$

O operador *SpatialConcatenation* (\oplus) é definido da seguinte forma:

$$\begin{aligned} D_{espacial} \oplus E_{espacial} &= \{D'_{espacial}\} \oplus \{E'_{espacial}\} \\ &= \{D'_{espacial}, E'_{espacial}\} \\ D_{espacial} \oplus M_{espacial} &= \{D'_{espacial}\} \oplus \{M'_{espacial}\} \\ &= \{D'_{espacial}, M'_{espacial}\} \\ M_{espacial} \oplus N_{espacial} &= \{M'_{espacial}\} \oplus \{N'_{espacial}\} \\ &= \{M'_{espacial}, N'_{espacial}\} \end{aligned}$$

A divisória “Camadas” é responsável por particionar a tabela por nível espacial, semelhante à divisória “Páginas”, que particiona a tabela de acordo com a combinação dos membros. Por exemplo, se a aplicação da álgebra tabular aos campos da divisória “Páginas” resultar no conjunto na forma normalizada $R = \{d_1e_1, \dots, d_1e_n, \dots, d_2e_1, \dots, d_2e_n, \dots, d_ne_1, \dots, d_ne_n\}$, onde d_i é o nome de determinado membro do campo qualitativo D e e_i é o nome de determinado membro do campo qualitativo E , uma tabela será gerada para cada elemento do conjunto R . O mesmo acontece para a divisória “Camadas”, ou seja, se o resultado da aplicação da álgebra tabular a essa divisória resultar no conjunto $S = \{D'_{espacial}, E'_{espacial}\}$, uma tabela será gerada para cada elemento

do conjunto S . Se, na divisória “Camadas”, os campos estiverem organizados de forma hierárquica, apenas o campo mais detalhado será utilizado como operando.

Na linguagem VizQL, cada célula da tabela contém um painel onde a porção dos dados referente àquela célula será exibida. Para permitir a visualização dos dados em mapas com sobreposição de camadas, um novo tipo de painel foi adicionado: o painel espacial. Assim, os painéis podem exibir os dados de forma convencional ou espacial.

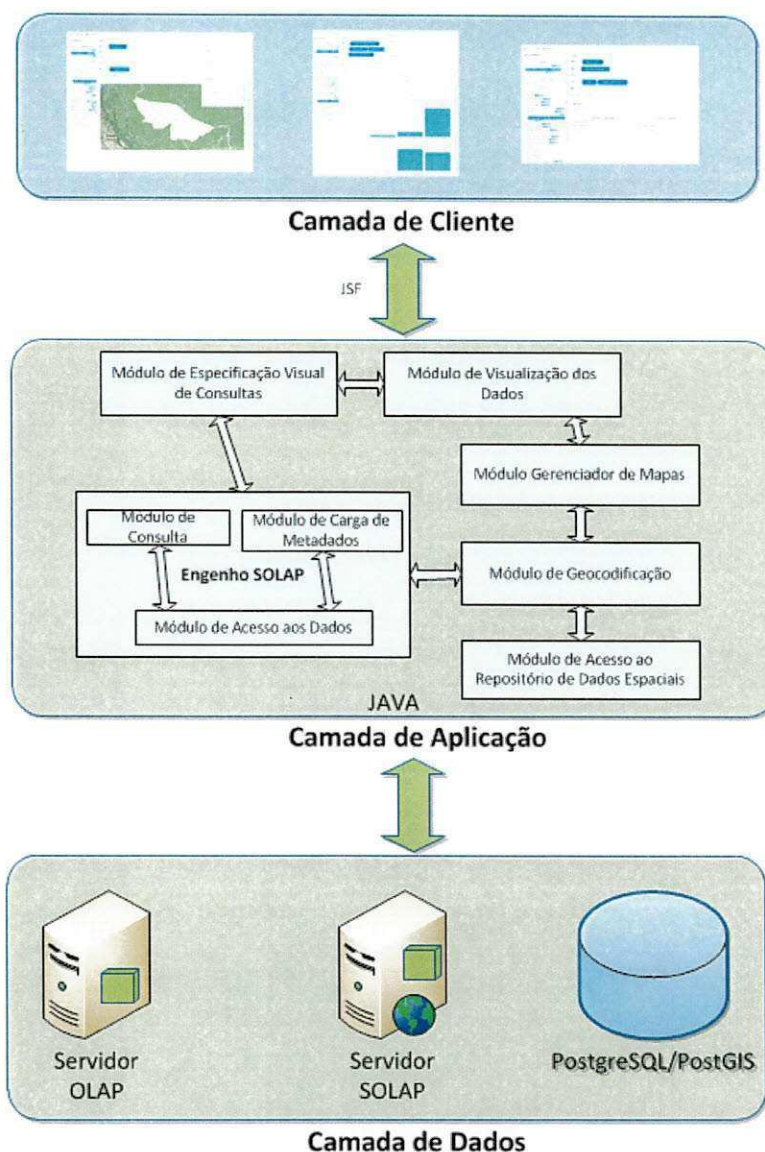
O painel espacial é resultado da sobreposição dos painéis das tabelas resultantes da álgebra tabular aplicada à divisória “Camadas”. Isso significa que, para exibir o resultado de uma consulta no mapa, as tabelas são sobrepostas e fundidas em uma só, diferentemente da divisória “Páginas”, que exibe uma tabela por vez. Outra diferença é que o painel convencional agrega os valores das medidas, enquanto que o painel espacial agrega os valores das medidas por geometria, também conhecida como feição.

Assim, as feições são agrupadas por campo. Cada conjunto gerado por meio desse agrupamento é uma camada, o que possibilitará a sobreposição de camadas, por exemplo, em um mapa. Quando níveis forem adicionados à divisória “Camadas”, a célula da tabela será composta de camadas. Além da divisória “Camadas”, restrições espaciais são especificadas para a linguagem VizQL. O OGC (*Open Geospatial Consortium*) [10] uniu esforços e desenvolveu padrões para modelar, acessar, armazenar e compartilhar dados geográficos. Os operadores topológicos definidos pelo OGC, com base no modelo DE-9IM (*Dimension Extended Nine-Intersection Model*), são *Touches*, *Within*, *Crosses*, *Overlaps*, *Disjoint*, *Contains*, *Equals*, *Intersects* e *Relate*. Esses operadores estão disponíveis para a especificação de filtros geográficos. O OGC também especifica os operadores *Buffer*, *ConvexHull*, *Envelop*, *Boundary*, *Centroid* e *PointOnSurface*. Eles estão disponíveis para uso nos filtros geométricos.

4.3. ARQUITETURA

A arquitetura do framework proposto nesta dissertação se divide em três camadas: a camada de cliente, a camada de aplicação e a camada de dados. A Figura 17 ilustra essa arquitetura.

Figura 17 – Arquitetura da solução SOLAP



Conforme mostrado na Figura 17, a Camada de Cliente compreende um conjunto de interfaces gráficas Web com as quais o usuário pode se conectar a um cubo espacial multidimensional, geocodificar membros, compor consultas por meio de uma especificação

visual e visualizar os dados. Para a construção das interfaces foram utilizadas as tecnologias *JavaServer Faces* (JSF) 2.0 [32] e *PrimeFaces* [33].

A camada de dados é composta pelas fontes de dados multidimensionais, de interesse de análise, e pelo repositório de dados espaciais. O *framework* é capaz de acessar vários servidores multidimensionais (servidores de cubos), em que é possível a utilização de diferentes tecnologias e fabricantes. O *framework* também dá suporte à geocodificação dos membros do cubo, possibilitando a análise espacial em cubos puramente OLAP. O repositório de dados da aplicação é mantido no SGBD PostgreSQL, com extensão espacial PostGIS. É nesse repositório que os dados espaciais resultantes da geocodificação dos membros do cubo são armazenados.

A camada de aplicação é responsável por implementar toda a lógica da aplicação e foi desenvolvida utilizando a tecnologia Java [34]. Ela dispõe de seis módulos: o de especificação visual de consultas, o de visualização dos dados, o gerenciador de mapas, o de acesso ao repositório de dados espaciais e o engenho SOLAP. Dentre esses módulos, destaca-se o engenho SOLAP, o principal módulo dessa camada, que provê a comunicação entre a aplicação e os servidores multidimensionais (OLAP ou SOLAP) acoplados na camada de dados. Todos os módulos são descritos a seguir.

4.3.1. Módulo de especificação visual de consultas

A especificação visual de consultas estimula o usuário a criar visualizações tabulares de forma rápida e incremental, abstraindo detalhes técnicos como linguagens de consultas. É composta de duas partes: a especificação da formatação da tabela e a especificação da visualização dos dados. O módulo de especificação visual de consultas controla o processo de execução e exibição das consultas e é responsável por transformar as interações dos usuários com a interface gráfica em objetos que compõem a especificação visual da consulta. As interações do usuário com a interface resultam em interações desse módulo com o engenho SOLAP e com o módulo de visualização dos dados.

A interface de especificação visual da consulta é dividida em duas janelas - uma posicionada à esquerda da tela e que contém os metadados do cubo, e outra à direita da tela, com as divisórias para a especificação visual da consulta (Figura 18). A janela esquerda organiza os metadados do cubo em dimensões e medidas. As dimensões são os campos qualitativos (categóricos), e as medidas, os campos quantitativos. Na janela direita, as

divisórias representam os eixos da consulta, onde os campos são adicionados para formular expressões VizQL.

Figura 18 – Interface para a especificação visual de uma consulta SOLAP

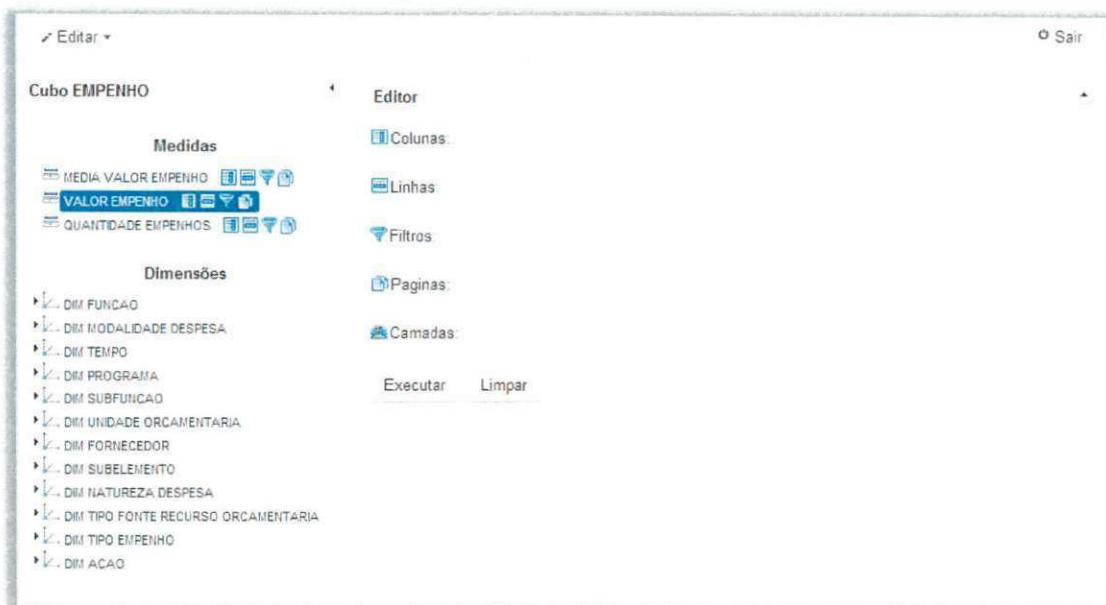
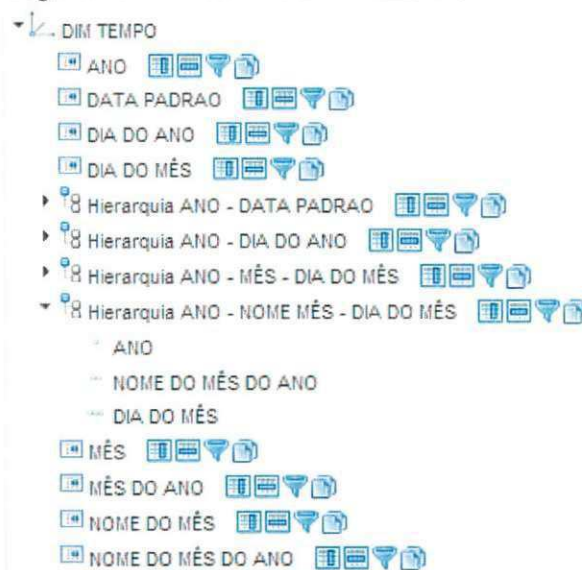









Figura 19 – Dimensão DIM TEMPO



Na janela dos metadados, para a Dimensão de Medidas, cada membro do Nível de Medidas será exibido ao usuário como um campo medida (📊). Para as demais dimensões, as hierarquias serão exibidas ao usuário da seguinte forma: as hierarquias que possuem apenas o nível All e o nível mais detalhado serão representadas apenas pelo nível mais detalhado como

um campo atributo de dimensão () , enquanto que as hierarquias com mais de um nível, além do All, serão representadas pelos seus níveis como um campo hierarquia () (Figura 19).

Os campos são adicionados às divisórias para especificar determinada consulta. Ao lado dos campos, na janela que contém os metadados do cubo, existem ícones representando cada divisória (Figura 19). As divisórias disponíveis são:  Colunas,  Linhas,  Páginas,  Filtros e  Camadas. Apenas os campos dos tipos atributo de dimensão e hierarquia podem ser adicionados às divisórias. Ao se adicionar um desses campos, o nível menos detalhado da hierarquia é adicionado à divisória, se a hierarquia possuir mais de um nível, ícones (+ e -) irão possibilitar a navegação nos demais níveis respeitando a ordem da hierarquia. Um nível só pode estar presente em uma única divisória.

As divisórias Colunas e Linhas formatam a estrutura da tabela; a divisória “Linhas” cria as linhas da tabela, e a divisória “Colunas” cria as colunas da tabela. Quando níveis são adicionados a uma dessas divisórias, os nomes dos membros são utilizados para criar os cabeçalhos da tabela, e quando uma medida é adicionada a uma dessas divisórias, o nome da medida é utilizado para criar os cabeçalhos da tabela. Quanto mais níveis lhes forem adicionados, mas detalhados serão os dados.

Por exemplo, ao adicionar a medida VALOR EMPENHO e o nível DESCRIÇÃO TIPO EMPENHO da dimensão DIM TIPO EMPENHO à divisória “Colunas” e o nível NOME MUNICÍPIO da dimensão DIM UNIDADE ORÇAMENTÁRIA à divisória “Linhas”, a tabela gerada apresentará o formato apresentado na Figura 20.

A divisória “Páginas” particiona os dados em páginas; cada página detém uma porção dos dados no formato tabular especificado pelo usuário (tabela). Quando um nível é adicionado à divisória “Páginas”, os nomes dos membros são utilizados para compor o cabeçalho da página, e se uma medida for adicionada, o nome da medida é utilizado para compor o cabeçalho da página. Para cada cabeçalho, uma página é gerada contendo uma tabela com a porção dos dados referentes ao cabeçalho. Se o nível MÊS da dimensão DIM TEMPO for adicionado à divisória “Páginas”, os dados serão particionados por mês, e cada mês terá uma tabela associada. Um componente para a navegação nas páginas deverá ser exibido. Um exemplo desse componente, que deve permitir que o usuário navegue selecionando a próxima página, a página anterior ou uma página específica, pode ser visto na Figura 21.

Figura 20 – Exemplo de tabela gerada pelas divisórias Colunas e Linhas

NOME MUNICÍPIO	ESTIMATIVO	GLOBAL	ORDINÁRIO
	VALOR EMPENHO	VALOR EMPENHO	VALOR EMPENHO
Assis Brasil			
Brasiléia			
Bujari			
Capixaba			
Cruzeiro do Sul			
Epitaciolândia			
Jordão			
Mâncio Lima			
Manoel Urbano			
Marechal Thaumaturgo			
Plácido de Castro			
Porto Acre			
Porto Walter			
Rio Branco			
Rodrigues Alves			
Santa Rosa do Purus			
Senador Guiomard			
Tarauacá			
Xapuri			

Figura 21 – Componente para a navegação nas páginas

Paginação

01/2012 >>>

01/2012

02/2012

03/2012

04/2012

05/2012

A divisória “Filtros” serve para selecionar uma determinada porção dos dados sem que seja necessário modificar a estrutura da tabela. Somente os dados relacionados aos membros presentes nessa divisória serão sumarizados e exibidos nas tabelas. Ao adicionar um nível à divisória “Filtros”, a interface irá exibir uma tela para que o usuário filtre os membros desse nível. Nenhum membro estará selecionado, ao contrário das demais divisórias onde, ao adicionar um nível, todos os membros são selecionados.

A Figura 22 exibe um exemplo em que o usuário adicionou o nível DESCRIÇÃO FUNÇÃO da dimensão DIM FUNÇÃO à divisória “Filtros”.

Figura 22 – Exemplo de tela para adição de um campo à divisória “Filtros”



A divisória “Camadas” foi adicionada à interface para que os níveis espaciais fossem exibidos nos mapas como camadas, e as camadas fossem sobrepostas. Só os níveis espaciais podem ser adicionados a essa divisória.

Nas divisórias, é possível filtrar os membros de um nível (Figura 23). O filtro pode ser convencional ou geográfico. No convencional, os membros são incluídos ou excluídos pelo usuário (Figura 24). No geográfico, o usuário irá especificar uma restrição espacial (topológica) para os membros (Figura 25). Outras dimensões espaciais podem ser utilizadas na criação das restrições.

Nas divisórias, é possível navegar nos níveis hierárquicos. Com apenas um clique do mouse no ícone de navegação (+ ou –) (Figura 23), podem-se executar *drill-down* e *roll-up*. *Drill-down* refere-se ao processo de visualizar os dados aumentando o nível de detalhe, enquanto que o *roll-up* é o processo por meio do qual se visualizam os dados diminuindo o nível de detalhe.

Se um nível da hierarquia exibe o ícone +, significa que os filhos dos seus membros ainda não estão sendo mostrados e que o usuário poderá realizar *drill-down* clicando no ícone.

Se um nível da hierarquia exibir o ícone –, significa que os filhos dos seus membros já estão sendo mostrados e que o usuário poderá realizar *roll-up* clicando no ícone.

Figura 23 – Divisórias para a especificação da formatação da tabela



Figura 24 – Filtro convencional para os membros

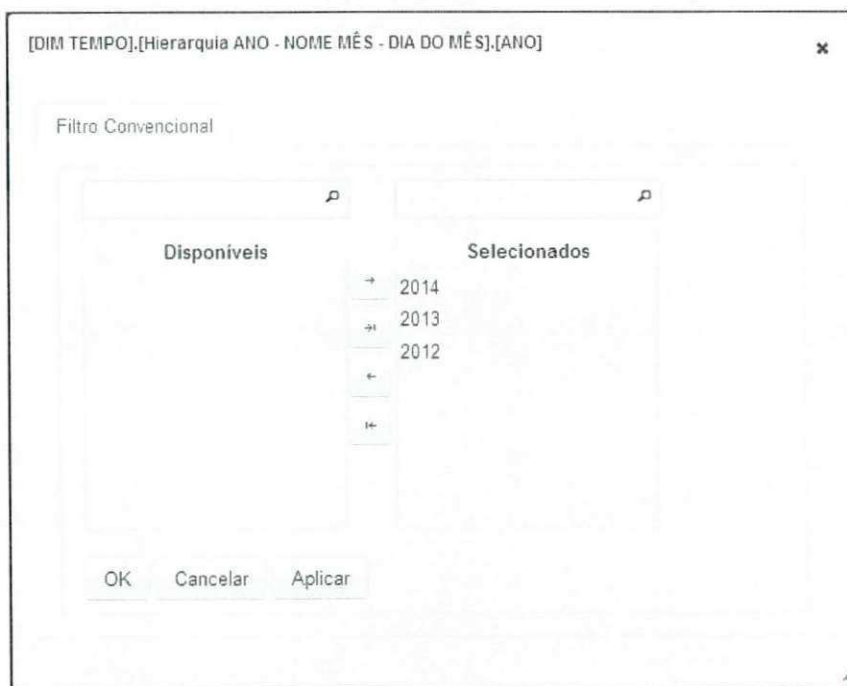


Figura 25 – Filtro geográfico para os membros

Na divisória “Camadas”, dos níveis utilizados de uma determinada hierarquia, apenas o mais detalhado será utilizado como camada para a exibição dos dados. O nível mais detalhado é destacado dos demais (Figura 26). Se os níveis pertencerem a diferentes hierarquias, serão sobrepostos. Por exemplo, ao se adicionar a hierarquia “ESTADO – MESORREGIÃO – MUNICÍPIO” à divisória “Camadas”, a princípio, o nível “NOME ESTADO” será exibido como camada para os dados. Ao se realizar *drill-down* no nível “NOME ESTADO”, o nível “NOME MESORREGIÃO” passará a ser utilizado como camada para os dados. Por fim, ao se realizar *drill-down* no nível “NOME MESORREGIÃO”, o nível “NOME MUNICÍPIO” passará a ser utilizado como camada para os dados.

Figura 26 – Divisória “Camadas”

Para formatar a visualização dos dados, estão disponíveis as marcações Texto, Gráfico e Legenda, que podem ser visualizadas em dois tipos de painel: convencional ou espacial. O tipo de painel determina a forma como as marcações são exibidas aos usuários.

Depois de definir a estrutura da tabela, o usuário deve especificar a forma como deve visualizar os dados informando, primeiro, o tipo de painel a ser utilizado (Figura 27).

Figura 27 – Componente para a escolha do tipo de painel



Ao definir o tipo de painel, um conjunto de tipos de marcação estará disponível para o usuário. O painel poderá ser formatado com os tipos de marcação disponíveis. A marcação do tipo Texto exibe o valor das medidas selecionadas. No tipo Legenda, as medidas são agrupadas em faixas; cada uma delas é representada por uma cor. Para cada medida, será gerada uma legenda, e apenas uma legenda será exibida por vez ao usuário. O número de faixa e a escala de cor são definidos pelo usuário (Figura 29). O critério para agrupar os valores da medida para as legendas pode ser Grupo ou Faixa de Valores. Para o critério Grupo de Valores, os valores da medida são ordenados de forma crescente e agrupados em um determinado número de classes, cada classe irá gerar uma faixa de valores. Para o critério Faixa de Valores, a amplitude dos valores será dividida de acordo com um determinado número de classes gerando faixas de valores que serão utilizadas para agrupar os valores das medidas.

No tipo Gráfico, as medidas são utilizadas para compor gráficos, os quais têm eixos. Ao configurar um gráfico, pelo menos um eixo deve conter uma medida, e aos demais eixos podem ser atribuídas outras medidas ou dimensões (Figura 28). Os tipos de gráfico disponíveis são: Barras, Linhas e Pizza.

Figura 28 – Especificação visual para o tipo de marcação Gráfico

The screenshot shows a dialog box titled 'Visualização' with a close button (x) in the top right corner. It contains two main sections: 'Painel' and 'Marcação'.
- In the 'Painel' section, there is a dropdown menu currently set to 'Convencional'.
- In the 'Marcação' section, there are five dropdown menus:
 - 'Marcação' is set to 'Gráfico'.
 - 'Tipo' is set to 'Barra'.
 - 'Eixo X' is set to 'DESCRIÇÃO FUNÇÃO'.
 - 'Eixo Y' is set to 'VALOR EMPENHO'.

Figura 29 – Especificação visual para o tipo de marcação Legenda

The screenshot shows a dialog box titled 'Visualização' with a close button (x) in the top right corner. It contains three main sections: 'Painel', 'Camadas', and 'Marcação'.
- In the 'Painel' section, there is a dropdown menu currently set to 'Espacial'.
- In the 'Camadas' section, there is a dropdown menu for 'Camada' set to 'NOME MESOREGIÕES'.
- In the 'Marcação' section, there are four dropdown menus:
 - 'Marcação' is set to 'Legenda'.
 - 'Medida' is set to 'VALOR EMPENHO'.
 - 'Rótulo' is set to 'NOME MESOREGIÕES'.
 - 'Grupos de valores' is set to '5'.
 - 'Faixa de valores' is currently empty.
- At the bottom, there is a 'Cor:' label followed by a small orange color swatch.

Quando adicionadas ao painel convencional, as marcações são exibidas nas células da tabela da seguinte maneira:

- Marcação Texto: o valor da medida é exibido em cada célula da tabela (Figura 30);
- Marcação Gráfico: um gráfico é exibido em cada célula da tabela (Figura 31);
- Marcação Legenda: a cor referente à faixa da legenda é aplicada ao texto contendo o valor da medida em cada célula da tabela.

Quando adicionadas ao painel Espacial, as marcações são exibidas nas células da tabela da seguinte maneira:

- Marcação Texto: em cada célula, o valor da medida é exibido sobre a geometria correspondente da camada;
- Marcação Gráfico: um gráfico é exibido sobre cada geometria da camada, para cada célula da tabela;
- Marcação Legenda: as cores referentes às faixas da legenda são aplicadas às geometrias da camada, onde a faixa correspondente a uma geometria é determinada pelo valor da medida selecionada (Figura 32).

Quando a marcação Legenda é especificada pelo usuário, um componente é exibido contendo a representação da legenda para cada camada.

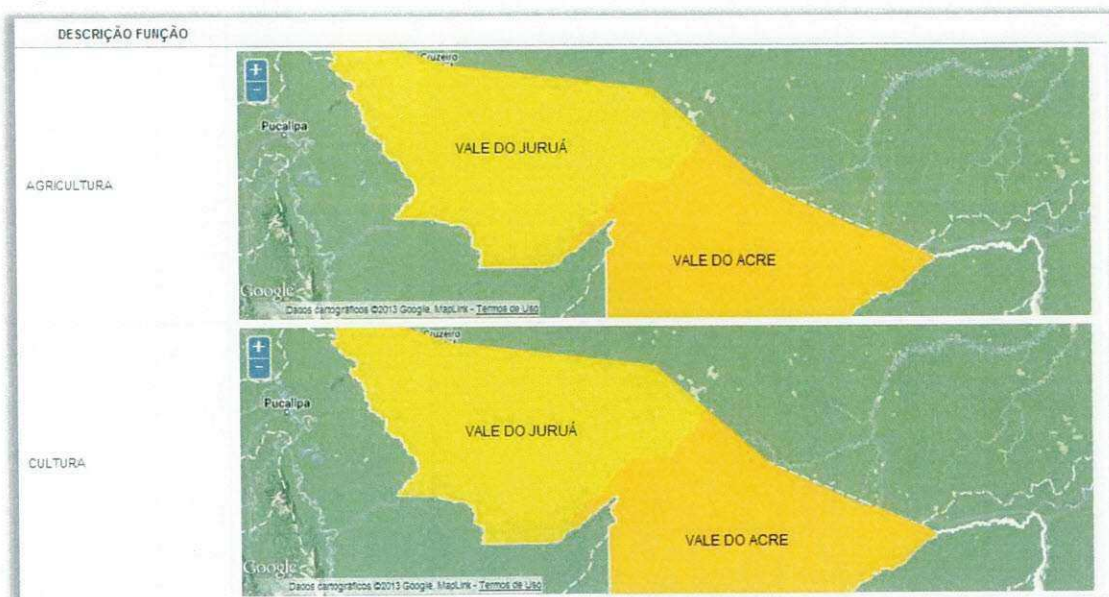
Figura 30 – Tabela marcada com texto

MÊS	MeasuresLevel	AGRICULTURA	CULTURA	ENERGIA	SAÚDE
01/2012	VALOR EMPENHO	RS 689144.76000	RS 410552.71000	RS 63570.00000	RS 16196638.65000
02/2012	VALOR EMPENHO	RS 918915.97000	RS 666093.65000	RS 79285.85000	RS 25039518.85000
03/2012	VALOR EMPENHO	RS 1642614.35000	RS 1043459.99000	RS 12638.50000	RS 17145606.28000

Figura 31 – Tabela marcada com gráfico resultante da especificação ilustrada na Figura 28



Figura 32 – Tabela marcada com legenda resultante da especificação ilustrada na Figura 29



4.3.2. Módulo de visualização dos dados

Ao receber o resultado de uma consulta visual, o módulo de especificação visual repassa o resultado e a marcação para o módulo de visualização dos dados para que os dados sejam transformados e apresentados no formato especificado. Dependendo do tipo de marcação, os dados precisam ser transformados, por exemplo, agrupados para gerar as camadas de um mapa ou os eixos de um gráfico. Assim, são transformados e depois repassados para o componente da interface adequado à sua visualização (e.g. tabelas com mapas, gráficos, texto e legenda). Os componentes da interface utilizados para a visualização dos dados são criados dinamicamente utilizando a suíte de componentes PrimeFaces.

4.3.3. Módulo gerenciador de mapas

Esse módulo é responsável por gerar os mapas para a visualização dos dados. Para exibir os dados em mapas, esse módulo deve receber do módulo de visualização dos dados um conjunto de dados espaciais e numéricos, originados de uma consulta, e a marcação. Uma vez gerados, os mapas serão exibidos nas células da tabela, originada da especificação visual. Um mapa é um objeto interativo, com o qual o usuário pode executar funções como *zoom in*, *zoom out* e *pan*. Para exibir os dados em mapas desenvolveu-se uma suíte de componentes para JSF utilizando a biblioteca OpenLayers [35].

O módulo de visualização dos dados interage com esse módulo ao receber o resultado de uma especificação visual que tenha uma marcação do tipo espacial. Os mapas também interagem com esse módulo ao realizar as funções *zoom* e *pan*.

4.3.4. Módulo de Acesso ao repositório de dados espaciais

Esse módulo foi implementado para recuperar os metadados e os dados das tabelas espaciais armazenadas no repositório de dados espaciais. Os metadados e os dados das tabelas espaciais são utilizados pelo módulo de geocodificação, cujo acesso é feito via *driver* JDBC [36] para o SGBD PostgreSQL com a extensão espacial PostGis.

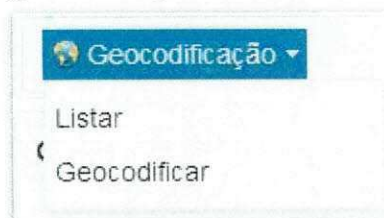
4.3.5. Módulo de geocodificação

O processo de associar coordenadas geográficas a determinado endereço é denominado geocodificação. O endereço é a descrição do local de interesse e pode ser representado de diversas formas: por meio de códigos (e.g. Código de Endereçamento Postal – CEP), de nomes (e.g. nome da cidade, nome do restaurante, etc.), combinação de nomes formando uma hierarquia (e.g. rua -> bairro -> cidade -> estado -> país), entre outras formas.

Caso o servidor OLAP não dê suporte a dados espaciais, é possível utilizar esse módulo para geocodificar os membros dos níveis, possibilitando a análise espacial dos cubos puramente OLAP. Essa estratégia permite ao usuário realizar a análise espacial em cubos OLAP sem que seja necessário o retrabalho de implementação desses cubos com o acréscimo das componentes espaciais utilizando tecnologias SOLAP. Porém, a geocodificação só é permitida em membros de dimensões, pois medidas não são suportadas. Cubos originados de servidores SOLAP não podem utilizar essa funcionalidade. Outra limitação é que os dados espaciais resultantes da geocodificação precisam ser importados para o repositório de dados espaciais do *framework*, a fim de permitir operações espaciais nesses dados. Dentre as funcionalidades desse módulo, está a de gerar e gerenciar mapeamentos entre registros do banco de dados espacial e membros do cubo multidimensional.

Assim, caso o servidor OLAP não dê suporte a dados espaciais, no *menu* da aplicação, estará disponível a opção geocodificação (Figura 33).

Figura 33 – Menu para a geocodificação



Para geocodificar os membros de um nível hierárquico, o usuário deve selecionar o subitem “Geocodificar” no menu “Geocodificação” da aplicação. Um passo a passo (*wizard*) para auxiliar o processo de geocodificação será exibido ao usuário. O *wizard* é composto de três abas que representam os passos a serem seguidos por ele.

Na primeira aba (Figura 34), o usuário deve selecionar a hierarquia em que deseja geocodificar os membros. Essa seleção é feita em três etapas: primeiro, seleciona-se a dimensão, depois, a hierarquia e, por fim, o nível. A propriedade padrão (*default*) utilizada na geocodificação é o nome do membro, cuja propriedade poderá ser alterada pelo usuário, caso deseje. Nessa aba, também é possível obter uma prévia dos valores da propriedade dos membros (Figura 35).

Na segunda aba (Figura 36), o usuário seleciona a tabela onde se encontram os dados espaciais para o processo de geocodificação; em seguida, seleciona a coluna que será utilizada para gerar o mapeamento entre membros do cubo e registros da tabela e, por fim, a coluna que contém as geometrias. Quando o usuário passa para a próxima aba, um mapeamento é gerado automaticamente e associa os valores da propriedade dos membros com os da coluna da tabela escolhida para a junção.

Na terceira aba (Figura 37), o mapeamento gerado é exibido ao usuário, que pode alterar o resultado do mapeamento que, uma vez salvo, será persistido no banco de dados. Para identificar o membro de forma unívoca, é persistido o nome único do membro no mapeamento, e para o registro de forma unívoca, o valor da chave primária do registro da tabela no mapeamento.

Figura 34 – Aba para a seleção do nível para o processo de geocodificação

A imagem mostra a interface de usuário 'Geocodificação' com a aba 'Propriedade do Nível' selecionada. O formulário contém os seguintes campos:

- Cubo:** EMPENHO
- Dimensão: *** DIM UNIDADE ORCAMENTARIA
- Hierarquia: *** NOME MUNICÍPIO
- Nível: *** NOME MUNICÍPIO

Existem dois ícones de configuração (uma engrenagem e um ícone de documento) à direita do campo 'Nível'. Uma caixa de texto 'Alterar a propriedade do membro.' aponta para o ícone de engrenagem. Outra caixa de texto 'Prévia dos valores da propriedade dos membros.' aponta para o ícone de documento. No rodapé da interface, há os botões 'Cancelar' e '→ Próximo'.

Figura 35 – Prévia dos valores da propriedade dos membros



Figura 36 – Aba para a seleção da tabela espacial



O nível em que os membros foram geocodificados é marcado como espacial para que o usuário possa identificá-lo e utilizá-lo na análise espacial. No painel que exibe os metadados do cubo, um ícone contendo um globo é exibido ao lado do nível que foi marcado como espacial.

Figura 37 – Aba com o resultado do processo de geocodificação

Geocodificação

Propriedade do Nível: Tabela Espacial: **Resultado**

Nível: NOVE MUNICÍPIO Tabela: PCM_MUNICIPIOS
 Propriedade: MEMBER_NAME Coluna de Junção: NM_MUNICIP

Mapeamento

MEMBER_NAME	NM_MUNICIP	
Acrelândia	ACRELÂNDIA	✎
Assis Brasil	ASSIS BRASIL	✎
Brasiléia	BRASILEIA	✎
Bujari	BUJARI	✎
Capixaba	CAPIXABA	✎
Cruzeiro do Sul	CRUZEIRO DO SUL	✎
Epitaciolândia	EPITACIOLÂNDIA	✎
Feijó	FEIJÓ	✓ ✎
Jordão	ACRELÂNDIA	✎
Manoel Urbano	ASSIS BRASIL	✎
Marechal Thaumaturgo	BRASILEIA	✎
Mâncio Lima	BUJARI	✎
Plácido de Castro	CAPIXABA	✎
Porto Alegre	CRUZEIRO DO SUL	✎
Porto Walter	EPITACIOLÂNDIA	✎
Rio Branco	FEIJÓ	✎
Rodrigues Alves	JORDÃO	✎
Santa Rosa do Purus	MÂNCIO LIMA	✎
Sena Madureira	MANOEL URBANO	✎
Senador Guiomard	MARECHAL THAUMATURGO	✎
Tarauacá	PLÁCIDO DE CASTRO	✎
Xapuri	PORTO WALTER	✎
	RIO BRANCO	✎
	RODRIGUES ALVES	✎
	SANTA ROSA DO PURUS	✎
	SENADOR GUIOMARD	✎
	SENA MADUREIRA	✎
	TARAUACÁ	✎
	XAPURI	✎

Cancelar Salvar

Anterior

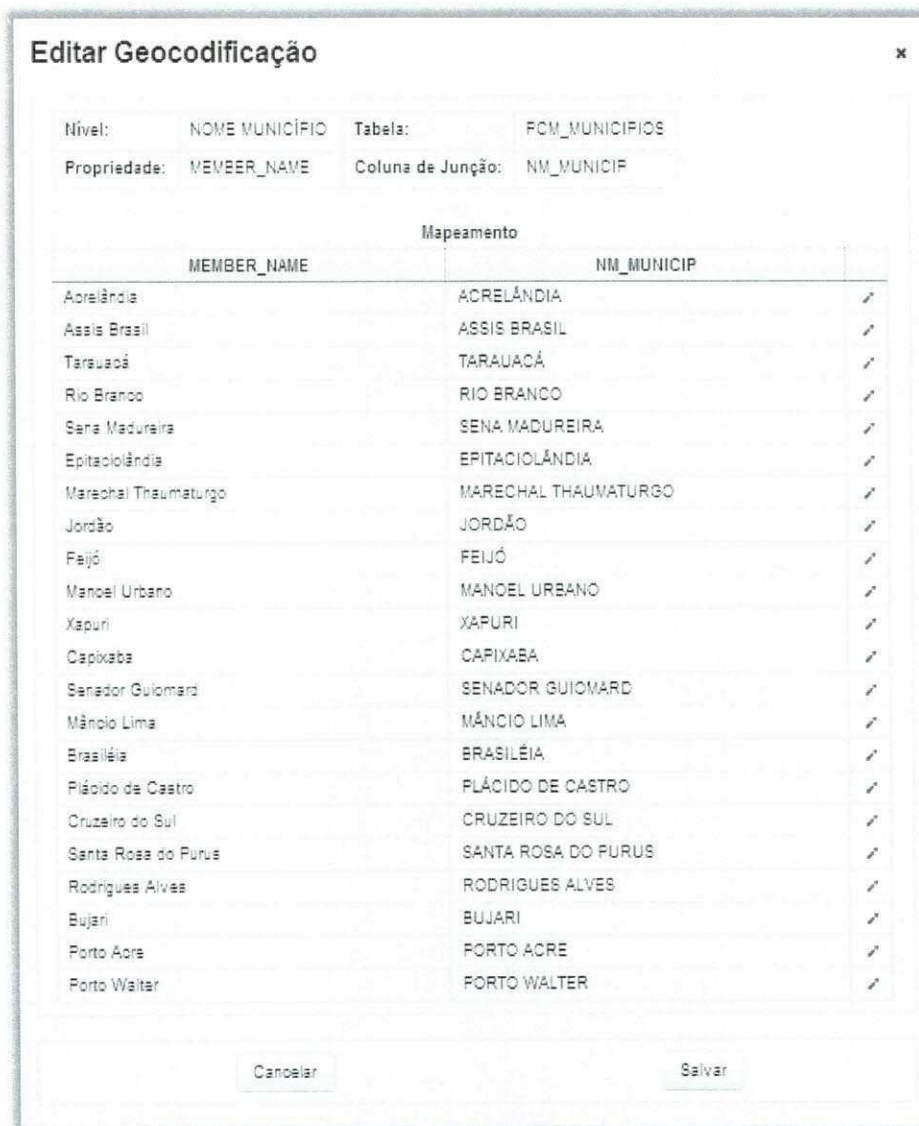
Para listar os níveis cujos membros foram geocodificados, o usuário deve escolher o subitem “Listar” no menu “Geocodificação” (Figura 33). Na tela onde os níveis geocodificados são listados, ele tem a opção de remover e editar o processo de geocodificação (Figura 38). Ao clicar no ícone “Editar”, a tela que contém o mapeamento gerado no processo

de geocodificação é exibida ao usuário. Nessa tela, o usuário pode alterar cada entrada do mapeamento (Figura 39).

Figura 38 – Tela listar níveis geocodificados



Figura 39 – Tela editar mapeamento da geocodificação



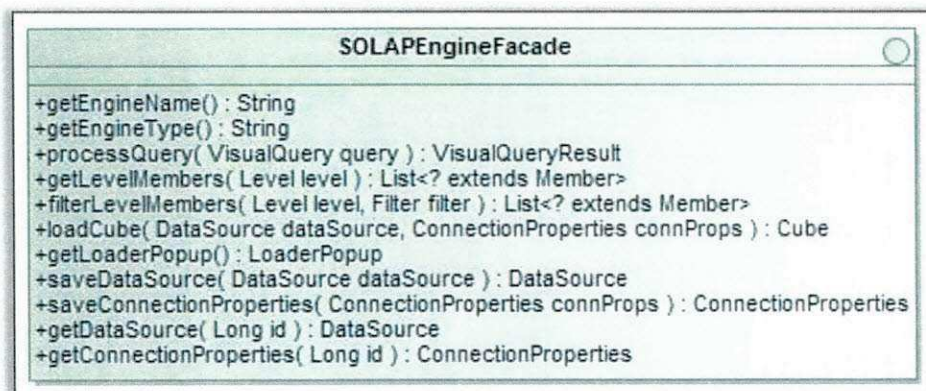
4.3.6. Engenho SOLAP

O engenho SOLAP divide-se em três módulos: módulo de acesso aos dados, módulo de carga dos metadados e módulo de consulta. Ele é responsável por:

- Permitir a conexão a determinada fonte de dados multidimensionais;
- Permitir a carga dos metadados dos cubos que se deseja analisar provenientes da fonte de dados multidimensionais a qual foi realizada a conexão;
- Traduzir a especificação visual na linguagem de consulta de destino;
- Executar a consulta de destino e recuperar os dados consultados.

A sua implementação depende do fabricante do servidor SOLAP a ser acessado, ou seja, para analisar cubos provenientes de um determinado servidor SOLAP, é necessário a implementação de um engenho capaz de realizar determinadas operações no servidor SOLAP necessárias ao funcionamento adequado da solução. O *framework* proposto dispõe de uma fachada para padronizar a implementação do engenho SOLAP e permitir a extensão da solução (Figura 40). Essa fachada será descrita com mais detalhe na seção 4.4, em que se abordam os pontos de extensão do *framework* proposto.

Figura 40 – Fachada para o engenho SOLAP



Módulo de acesso aos dados

O módulo de acesso aos dados possibilita ao usuário se conectar à fonte de dados multidimensionais e escolher o cubo que deseja analisar. Também é de responsabilidade desse módulo executar consultas na linguagem da tecnologia acessada e retornar os resultados dessas consultas. Para realizar o acesso aos dados, é necessário informar as propriedades de conexão que se classificam como propriedades da fonte e propriedades do cubo. As propriedades da fonte informam onde e como se conectar à fonte de dados multidimensionais, enquanto que as propriedades do cubo informam qual cubo que pertencente à fonte deve ser acessado. Objetos foram modelados com esse propósito: *ConnectionProperties* e *DataSource*.

O módulo de carga dos metadados e o módulo de consulta do engenho SOLAP interagem com esse módulo, que é especialista, ou seja, sua implementação depende da tecnologia a ser acessada e detém o conhecimento de como manipular esse tipo de fonte.

Módulo de carga dos metadados

O módulo de carga dos metadados é responsável pela conexão e pela recuperação dos metadados do cubo, por meio do módulo de acesso aos dados. Os metadados originários da fonte de dados multidimensionais são convertidos em objetos, que são carregados em memória para ser utilizados posteriormente pelos demais módulos do sistema. Algumas classes foram modeladas para esse propósito: *Cube*, *Dimension*, *Hierarchy*, *Level*, *Measure* e *Property*. O diagrama de classe está ilustrado na Figura 15.

Para a recuperação dos metadados, os objetos *ConnectionProperties* e *DataSource* são informados, e um objeto *Cube* é recuperado como resultado. Assim, para realizar a análise dos dados, o usuário deve se conectar ao servidor SOLAP e escolher o cubo que deseja analisar. Depois de feita a conexão ao cubo, seus metadados são carregados a fim de serem utilizados posteriormente por outros módulos do sistema. Na primeira vez em que o acesso é realizado, o usuário tem a opção de salvar as propriedades de conexão ao cubo para que ele seja acessado posteriormente. Ao salvar os dados da conexão do cubo, as propriedades de conexão da fonte de dados também são salvas.

A Figura 41 exibe a tela de gerenciamento dos cubos, na qual são listadas as fontes de dados e os cubos com os quais o usuário já se conectou. Ele também pode criar uma nova conexão ou remover uma já existente.

Figura 41 – Tela de gerenciamento dos cubos

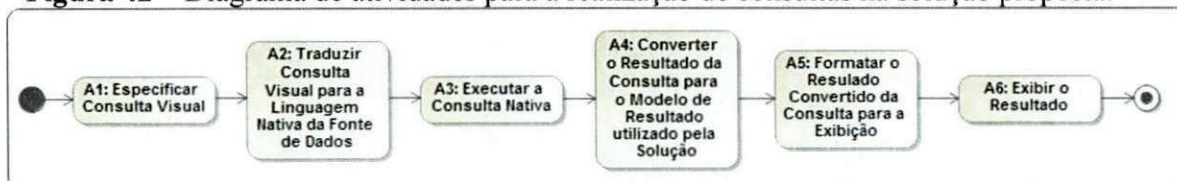


A conexão a um cubo deve ser feita por meio de uma fonte de dados já existente (salva previamente) ou de uma nova conexão. Para criar uma nova conexão, o usuário deve escolher um dos tipos de engenho SOLAP adequado à conexão à fonte de dados requerida. O engenho SOLAP irá disponibilizar um *popup* onde as informações necessárias à conexão devem ser informadas. Para cada tecnologia de fonte de dados multidimensionais, deve ser implementado e disponibilizado um engenho SOLAP para a conexão e análise dos cubos. Um engenho para acesso aos cubos via protocolo *XML for Analysis* (XMLA) [37] foi implementado e disponibilizado (seção 4.4.3).

Módulo de consulta

O módulo de consulta é responsável por traduzir as consultas visuais em consultas na linguagem da tecnologia de destino, de executá-las, por meio do módulo de acesso aos dados, e de retornar os resultados das consultas no modelo utilizado pela aplicação. O diagrama de atividades ilustrado na Figura 42 exibe as atividades necessárias para a realização de consultas na solução proposta, sendo que as atividades A2, A3 e A4 são de responsabilidade do engenho SOLAP, especificamente do módulo de consulta.

Figura 42 – Diagrama de atividades para a realização de consultas na solução proposta.



Para recuperar os dados, além das propriedades de conexão, uma consulta visual, representada pelo objeto *VisualQueryAdapter*, é passada como parâmetro para esse módulo. A consulta visual é então traduzida e executada. O processo de tradução da consulta visual para a linguagem da tecnologia de destino é exemplificado na seção 4.4.3. Um objeto que implementa a interface *VisualQueryResult* é retornado como resultado de uma consulta.

A consulta visual é composta de divisórias, que serão chamadas de eixos (*Axis*) nos objetos. Os eixos são compostos de hierarquias. Cada nível da hierarquia está associado um estado de nível, que mantém informações sobre quais membros estão selecionados, se o nível está sendo usado e se está detalhado (*drilled*). O diagrama de classe para o objeto *VisualQuery* é ilustrado na Figura 43.

Figura 43 – Diagrama de classe para a consulta visual

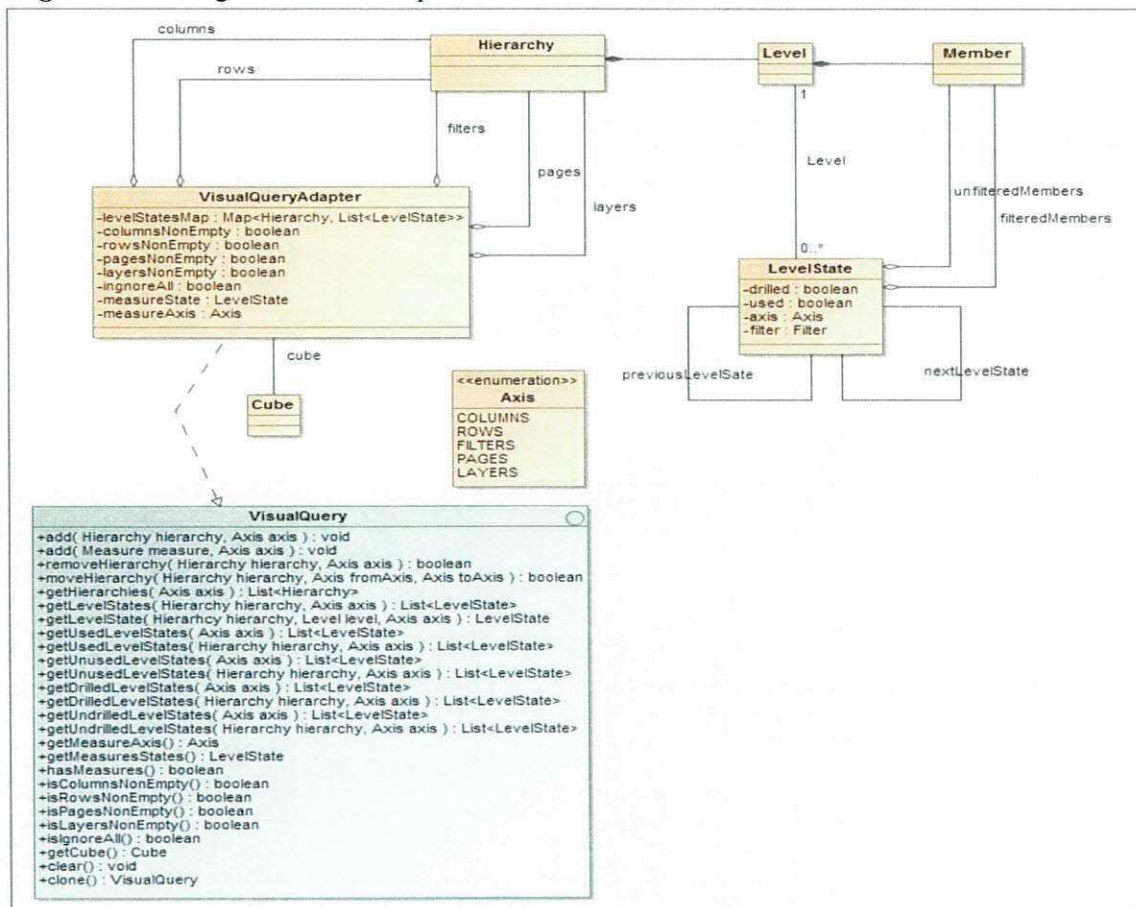
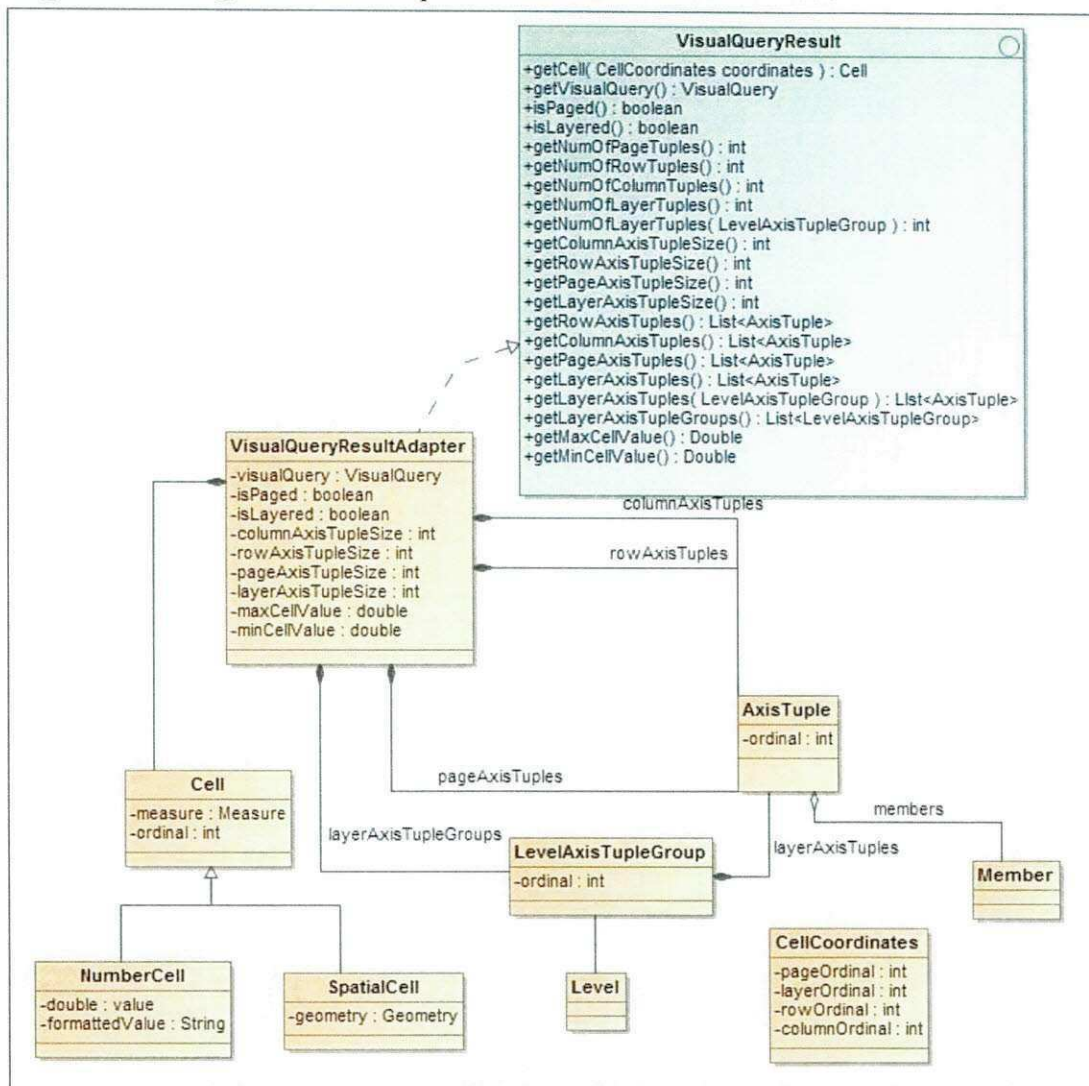


Figura 44 – Diagrama de classe para o resultado da consulta visual



O resultado de uma consulta visual é composto por células, tuplas e grupos (Figura 44). Para os eixos Colunas, Linhas e Páginas da consulta visual, o resultado da consulta retorna uma lista de *tuplas*. Para o eixo Camadas, o resultado da consulta retorna *tuplas* agrupadas em grupos. O eixo Filtro da consulta visual só serve para filtrar o conteúdo das células.

Uma *tupla* é composta por membros e contém um atributo (*ordinal*) que indica sua posição no eixo. A *tupla* tem uma dimensionalidade, ou seja, para cada nível utilizado no eixo da consulta visual, ela terá um membro representando o nível, respeitando a ordem de inserção dos níveis nos eixos.

Figura 45 – Cubo para os exemplos

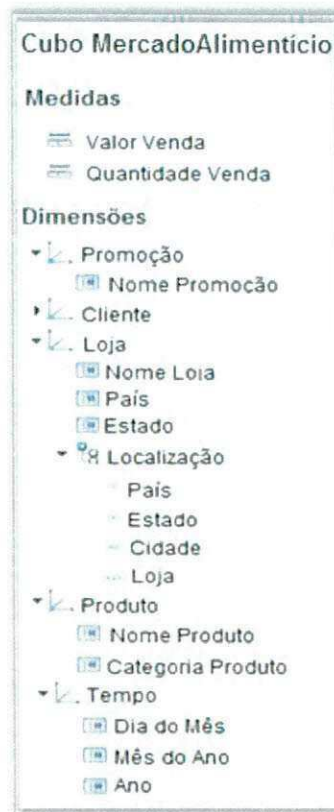
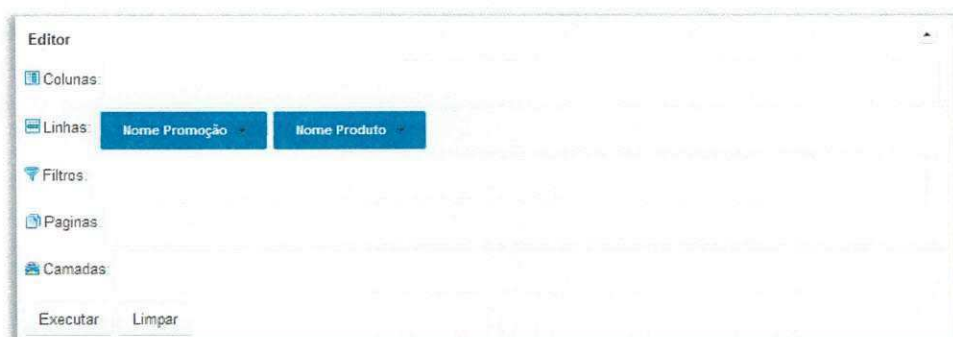


Figura 46 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas”



Por exemplo, considerando o cubo apresentado na Figura 45, os níveis “Nome Promoção” e “Nome Produto” foram adicionados ao eixo “Linhas” da consulta visual nessa ordem (Figura 46); para o nível “Nome Promoção”, selecionaram-se os membros “Super Desconto” e “Cupom Promocional”; e para o nível “Nome Produto”, os membros “Café” e “Biscoito”. Logo, utilizando a notação JSON, as possíveis *tuplas* para o eixo “Linhas” são:

Código 1 – JSON com as tuplas para o eixo "Linhas"

```
{ordinal:0, membros:["Super Desconto", "Café"]}
{ordinal:1, membros:["Super Desconto", "Biscoito"]}
{ordinal:2, membros:["Cupom Promocional", "Café"]}
{ordinal:3, membros:["Cupom Promocional", "Biscoito"]}
```

A interseção de *tuplas* provenientes dos eixos, onde cada *tupla* pertence a um eixo diferente, identifica uma célula contendo o valor de determinada medida. A célula é localizada por meio de uma coordenada que contém os ordinais (posições) das *tuplas* na seguinte ordem: Páginas, Linhas, Colunas e Camadas.

Figura 47 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas” e o nível de medidas no eixo “Colunas”



Utilizando o exemplo ilustrado na Figura 46, ao adicionar o nível “Medidas” ao eixo “Colunas” e selecionando a medida “Valor Venda” (Figura 47), o resultado da consulta é este:

Código 2 – Código JSON contendo o resultado do exemplo ilustrado na Figura 47

```
{pagesAxisTuples[
  {ordinal:0,membros:["Default"]}
],
rowsAxisTuples:[
  {ordinal:0, membros:["Super Desconto", "Café"]},
  {ordinal:1, membros:["Super Desconto", "Biscoito"]},
  {ordinal:2, membros:["Cupom Promocional", "Café"]},
  {ordinal:3, membros:["Cupom Promocional", "Biscoito"]}
],
columnsAxisTuples:[
  {ordinal:0,membros:["Valor Venda"]}
]}
```

```

],
layersAxisGroups: [
  {level: "Default", layersAxisTuples: [
    {ordinal:0, members:["Default"]}
  ]},
  cells: {
    "0,0,0,0":
    {value: 191940.0, formattedValue: "R$ 191.940,00"},
    "0,1,0,0":
    {value: 24597.0, formattedValue: "R$ 24.597,00"},
    "0,2,0,0":
    {value: 94814.0, formattedValue: "R$ 94.814,00"},
    "0,3,0,0":
    {value: 12202.0, formattedValue: "R$ 12.202,00"}}
  }
}

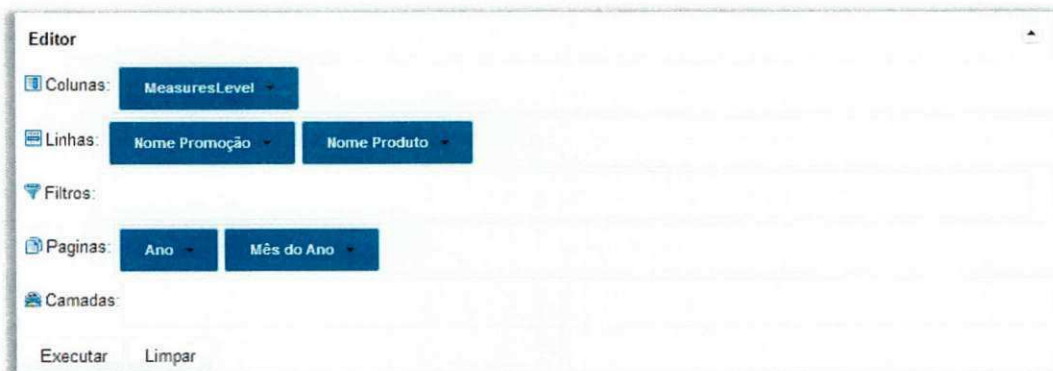
```

Mesmo não adicionando níveis aos eixos, uma *tupla* contendo o valor “Default” é retornada. A Tabela 2 pode ser gerada com este resultado:

Tabela 2 – Resultado consulta visual

Nome Promoção	Nome Produto	Valor Venda
Super Desconto	Café	R\$ 191.940,00
Super Desconto	Biscoito	R\$ 24.597,00
Cupom Promocional	Café	R\$ 94.814,00
Cupom Promocional	Biscoito	R\$ 12.202,00

Figura 48 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas”, o nível de medidas no eixo “Colunas” e os níveis “Ano” e “Mês do Ano” no eixo “Páginas”



Para o eixo Páginas, cada *tupla* corresponde a uma porção dos dados, denominada de página. Por exemplo, adicionando os níveis “Ano” e “Mês do Ano” ao eixo “Páginas” (Figura

48) e selecionando o membro “2012” do nível “Ano” e os membros “Janeiro” e “Fevereiro” do nível “Mês do Ano”, utilizando o exemplo anterior, será gerado o seguinte resultado:

Código 3 – Código JSON contendo o resultado do exemplo ilustrado na Figura 48

```

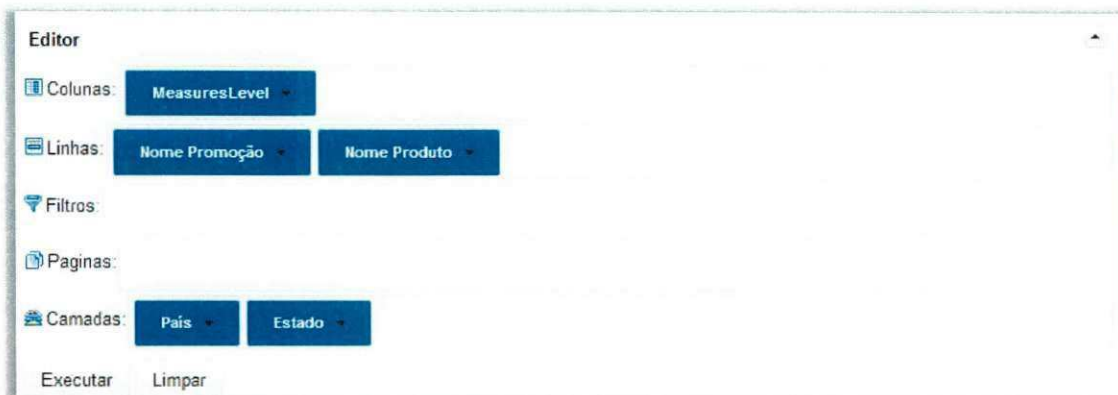
{pagesAxisTuples:[
  {ordinal:0, membros:["2012", "Janeiro"]},
  {ordinal:1, membros:["2012", "Fevereiro"]}
],
rowsAxisTuples:[
  {ordinal:0, membros:["Super Desconto", "Café"]},
  {ordinal:1, membros:["Super Desconto", "Biscoito"]},
  {ordinal:2, membros:["Cupom Promocional", "Café"]},
  {ordinal:3, membros:["Cupom Promocional", "Biscoito"]}
],
columnsAxisTuples:[
  {ordinal:0, membros:["Valor Venda"]}
],

layersAxisGroups:[
  {level: "Default", layersAxisTuples:[
    {ordinal:0, membros:["Default"]}
  ]},
  {cells:[
    "0,0,0,0":
    {value: 191940.0, formattedValue: "R$ 52.320,00"},
    "0,1,0,0":
    {value: 24597.0, formattedValue: "R$ 12.250,00"},
    "0,2,0,0":
    {value: 94814.0, formattedValue: "R$ 40.354,00"},
    "0,3,0,0":
    {value: 12202.0, formattedValue: "R$ 3.202,00"},
    "1,0,0,0":
    {value: 191940.0, formattedValue: "R$ 15.940,00"},
    "1,1,0,0":
    {value: 24597.0, formattedValue: "R$ 9.597,00"},
    "1,2,0,0":
    {value: 94814.0, formattedValue: "R$ 17.814,00"},
    "1,3,0,0":
    {value: 12202.0, formattedValue: "R$ 2.202,00"}]}]}

```

O resultado ilustrado no Código 3 tem duas páginas: a primeira referente a “2012, Janeiro”, contendo as quatro primeiras células do resultado, e a segunda, a “2012, Fevereiro”, contendo as demais células.

Figura 49 – Exemplo com os níveis “Nome Promoção” e “Nome Produto” no eixo “Linhas”, o nível de medidas no eixo “Colunas” e os níveis “País” e “Estado” no eixo “Camadas”



Para o eixo “Camadas”, cada nível é representado por um grupo. Esse grupo é composto por *tuplas*, cada uma das quais corresponde a um membro do nível relacionado ao grupo. Cada *tupla* de um grupo é um critério de agregação dos dados. Nesse eixo, somente o nível mais detalhado de uma hierarquia é utilizado para agregar os dados. No exemplo ilustrado na Figura 48, removendo os níveis do eixo “Páginas” e adicionando o nível “País” da hierarquia “País” e o nível “Estado” da hierarquia “Estado” ao eixo “Camadas” (Figura 49), selecionando o membro “Brasil” para o nível “País” e os membros “Paraíba” e “Bahia” para o nível “Estado”, o resultado será:

Código 4 – Código JSON contendo o resultado do exemplo ilustrado na Figura 49

```
{pagesAxisTuples[
  {ordinal:0,members:["Default"]}
],
columnsAxisTuples:[
  {ordinal:0,members:["Valor Venda"]}
],
rowsAxisTuples:[
  {ordinal:0, membros:["Super Desconto", "Café"]},
  {ordinal:1, membros:["Super Desconto", "Biscoito"]},
  {ordinal:2, membros:["Cupom Promocional", "Café"]},
  {ordinal:3, membros:["Cupom Promocional", "Biscoito"]}
],
layersAxisGroups:[
  {level: "País", layersAxisTuples:[
    {ordinal:0, membros:["Brasil"]}
  ]},
  {level: "Estado", layersAxisTuples:[
    {ordinal:1, membros:["Paraíba"]},
    {ordinal:2, membros:["Bahia"]}
  ]}
]}
```

```

    ]}],
    cells:[
      "0,0,0,0":
      {value: 191940.0, formattedValue: "R$ 191.940,00"},
      "0,0,0,1":
      {value: 24597.0, formattedValue: "R$ 24.597,00"},
      "0,0,0,2":
      {value: 94814.0, formattedValue: "R$ 94.814,00"},
      "0,1,0,0":
      {value: 12202.0, formattedValue: "R$ 12.202,00"},
      "0,1,0,1":
      {value: 191940.0, formattedValue: "R$ 52.320,00"},
      "0,1,0,2":
      {value: 24597.0, formattedValue: "R$ 12.250,00"},
      "0,2,0,0":
      {value: 94814.0, formattedValue: "R$ 40.354,00"},
      "0,2,0,1":
      {value: 12202.0, formattedValue: "R$ 3.202,00"},
      "0,2,0,2":
      {value: 191940.0, formattedValue: "R$ 15.940,00"},
      "0,3,0,0":
      {value: 24597.0, formattedValue: "R$ 9.597,00"},
      "0,3,0,1":
      {value: 94814.0, formattedValue: "R$ 17.814,00"},
      "0,3,0,2":
      {value: 12202.0, formattedValue: "R$ 2.202,00"}]}
  ]}],

```

No eixo “Camadas”, as *tuplas* resultantes são agrupadas por nível. Essa formatação permite a sobreposição de camadas.

4.4. PONTOS DE EXTENSÃO DA SOLUÇÃO

Para que a solução possa realizar a análise de cubos de dados provenientes de determinada tecnologia, é necessário estender o engenho SOLAP para que ele se torne capaz de tratar detalhes de implementação dessa tecnologia-alvo.

Para possibilitar a extensão do engenho, foram definidas interfaces de comunicação. O processo de extensão do *framework* proposto é detalhado nas seções seguintes.

4.4.1. Fachada para o engenho SOLAP

O principal ponto para a extensão do *framework* proposto é a implementação da fachada do engenho SOLAP, que é responsável pela comunicação entre a solução proposta e a fonte de dados multidimensional. As trocas de mensagens entre a solução e o engenho SOLAP consistem, basicamente, em solicitar metadados e dados de um cubo. A fachada

apresentada na Figura 40 padroniza essa troca de mensagens. Para solicitar os metadados de um cubo ao engenho SOLAP, a fachada dispõe do método denominado *loadCube*, que recebe como parâmetros as propriedades de conexão da fonte de dados e do cubo e retorna como resultado um objeto que representa o cubo (Figura 15).

Para solicitar dados de um cubo ao engenho SOLAP, a fachada dispõe de três métodos: *processQuery*, *getLevelMembers* e *filterLevelMembers*. O método *processQuery* recebe como parâmetro um objeto que modela a consulta visual (Figura 43), o qual é parte da especificação visual definida pelo usuário. Essa consulta visual deve ser traduzida para a linguagem de consulta da tecnologia de origem e executada. O resultado proveniente da execução da consulta deve ser modelado em um objeto de retorno denominado *VisualQueryResult* (Figura 44).

O método *getLevelMembers* recebe como parâmetro um objeto que representa um nível hierárquico. Esse nível deve ser utilizado para recuperar os membros do cubo, que são retornados. Por fim, o método *filterLevelMembers* recebe como parâmetro, além do nível, um filtro que pode ser convencional ou espacial (Figura 24 e Figura 25). Esse filtro será utilizado para selecionar os membros que devem ser recuperados.

Ao implementar a fachada, a solução identificará, automaticamente, essa implementação, por meio dos serviços de Contextos e Injeção de Dependência (*Contexts and Dependency Injection - CDI*) presentes na plataforma Java EE [38], e a registrará para uso. Um nome e um tipo devem ser associados ao engenho SOLAP. O tipo é utilizado pelo gerenciador de engenhos de BI, que mantém o mapeamento entre tipos e implementações disponíveis na solução. O nome é utilizado pela interface gráfica para ser apresentado ao usuário (Figura 41).

4.4.2. Interface para as propriedades de conexão

O processo de comunicação requer do usuário que ele informe as propriedades de conexão. Essa informação será utilizada toda vez que o engenho SOLAP precise se comunicar com a fonte de dados. Logo, outro ponto de extensão é a implementação dessa interface com o usuário.

As propriedades de conexão dependem da tecnologia a ser acessada. Assim, os campos a serem informados variam de acordo com a tecnologia. Para que o usuário possa

informar os campos necessários à conexão, um componente Web deve ser implementado utilizando o modelo (*template*) exibido na Figura 50.

A fachada para o engenho SOLAP dispõe de um método denominado *getLoaderPopup*, que retorna um objeto denominado *LoaderPoup*, que contém as informações necessárias para a exibição do componente. Esse objeto é utilizado pela interface, que lista todos os engenhos disponíveis na solução (Figura 41) e deve ser implementado. Ao clicar no nome do engenho que o usuário deseja utilizar para se conectar a um cubo, é exibida uma janela (*popup*) com o componente.

Figura 50 – *Template* do componente para informar as propriedades da conexão

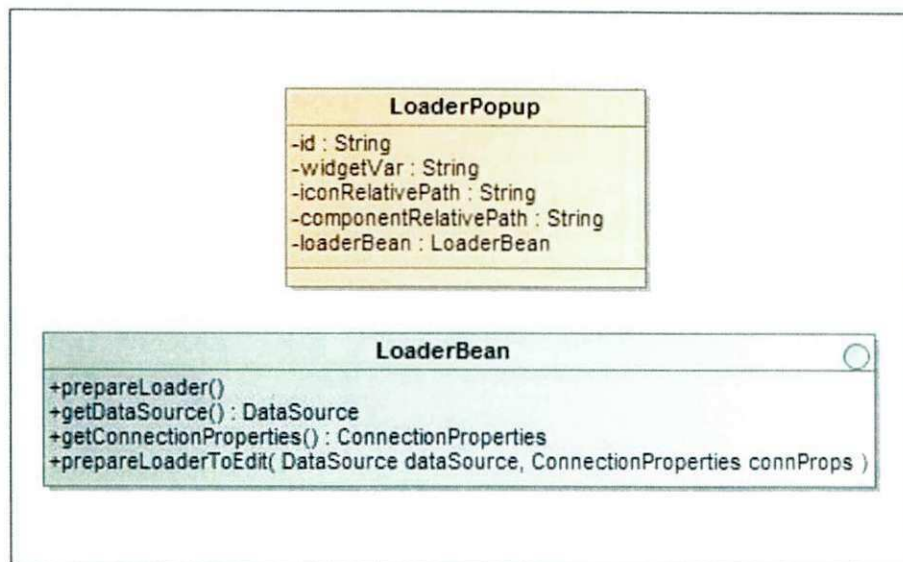
```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3
4 <html xmlns="http://www.w3.org/1999/xhtml"
5     xmlns:h="http://java.sun.com/jsf/html"
6     xmlns:f="http://java.sun.com/jsf/core"
7     xmlns:p="http://primefaces.org/ui"
8     xmlns:ui="http://java.sun.com/jsf/facelets">
9
10
11 <p:dialog header="" modal="true" id="" widgetVar="">
12
13   <h:form>
14
15     <!-- Your Code Here! -->
16
17     <p:commandButton value="Conectar" action=""
18       oncomplete="saveCubeConfirmation.show();" />
19
20   </h:form>
21 </p:dialog>
22
23 </html>

```

O objeto *LoaderPopup* é composto por outro objeto denominado *LoaderBean* (Figura 51), que também deve ser implementado. O *LoaderBean* é o controlador responsável por preparar o componente para ser exibido e por permitir o acesso às propriedades de conexões criadas pelo usuário. Depois de informar as propriedades e realizar a conexão, o controle da aplicação é devolvido à solução por meio do fragmento de código exibido na Figura 50, especificamente nas linhas 17 e 18. Nesse momento, o usuário é questionado sobre se deseja salvar as propriedades da conexão.

Figura 51 – Diagrama de classes para os objetos LoaderPopup e LoaderBean



Caso o usuário opte por salvar as propriedades da conexão, os métodos *saveDataSource* e *saveConnectionProperties* da fachada serão utilizados.

A especificação *Java Persistence Annotations* (JPA) [39] é utilizado para persistir os objetos e facilitar a extensão da solução. Assim, para que os objetos que estendem as classes *DataSource* e *ConnectionProperties* possam ser persistidos, basta mapear os atributos das classes utilizando as anotações da especificação JPA. Um exemplo de uma classe que estende a classe *DataSource* pode ser observado na Figura 52. Nele, os atributos da classe *JDBCDataSource* são mapeados com as anotações da especificação JPA.

Uma classe genérica que implementa o paradigma DAO (*Data Access Object*) capaz de realizar CRUD (*Create, Retrieve, Update and Delete*) de objetos foi implementada utilizando-se o conceito *Generics* disponível na linguagem Java. A classe *GenericDAOImpl* implementa a interface *GenericDAO* e disponibiliza métodos genéricos para a realização do CRUD de objetos de determinado tipo informado na definição da classe DAO (Figura 53).

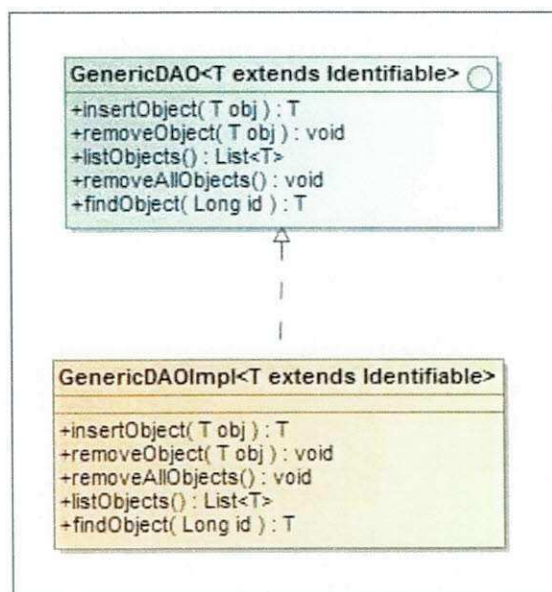
Figura 52 – Classe JDBCDataSource

```

1 package br.gov.ac.tce.solap.modelo.jdbc;
2
3 import javax.persistence.Column;
4
5 @Entity
6 @Table(name = "JDBC_DATA_SOURCE")
7 @Inheritance(strategy = InheritanceType.JOINED)
8 @PrimaryKeyJoinColumn(name="ID_DATA_SOURCE")
9 public abstract class JDBCDataSource extends DataSource{
10
11     @Column(name = "USUARIO")
12     protected String username;
13
14     @Column(name = "SENHA")
15     protected String password;
16
17     @Column(name = "HOST")
18     protected String host;
19
20     @Column(name = "PORTA")
21     protected String port;
22
23     @Column(name = "DRIVER")
24     protected String driver;
25
26     @Transient
27     protected String defaultPort;
28
29     public JDBCDataSource(){
30         super();
31     }
32
33     public abstract String getUrl(JDBCConnectionProperties connProperties) throws ConnectionException;

```

Figura 53 – Diagrama de classes para o GenericDAO




4.4.3. Engenho SOLAP para servidores XMLA

Focado na proposta de um *framework* para a análise SOLAP de várias fontes de dados multidimensionais, desenvolveu-se também, neste trabalho de dissertação, um engenho SOLAP para servidores que disponibilizam seus dados por meio do protocolo *XML for Analysis* (XMLA) [37]. Nesta seção, alguns aspectos importantes relacionados à implementação do engenho para servidores XMLA serão apresentados.

Na implementação do engenho SOLAP para XMLA, foi utilizado o *driver* XMLA disponibilizado pela API do *Open Java API for OLAP* (olap4j) [40], que também é uma especificação aberta para a construção de aplicações OLAP baseada na especificação JDBC [36]. Esse engenho disponibiliza um *popup* que foi implementado utilizando-se o modelo (*template*) exibido na Figura 50, onde as informações necessárias à conexão devem ser informadas (Figura 54).

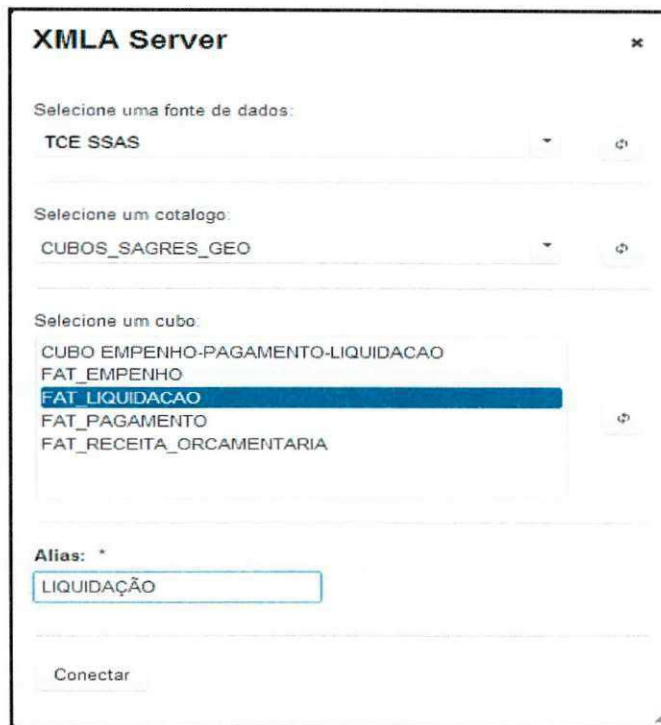
Figura 54 – *Popup* para conexão à fonte de dados



The image shows a window titled "XMLA Server" with a close button (x) in the top right corner. The window contains the following fields and labels:

- A label "Atribua um nome para a fonte de dados: *" followed by a text input field.
- A label "Insira o URL do servidor: *" followed by a text input field.
- A label "Insira as informações de autenticação do banco de dados:" followed by two text input fields:
 - "Nome de usuário:"
 - "Senha:"
- A "Conectar" button at the bottom left.

Uma vez conectado à fonte de dados, o usuário deve escolher qual cubo deseja analisar. Depois, deve informar um apelido (aliás) para o cubo, que será utilizado para identificar o cubo no sistema (Figura 55).

Figura 55 – Tela para a escolha do cubo

The screenshot shows a web application window titled "XMLA Server". It contains three main sections for configuration:

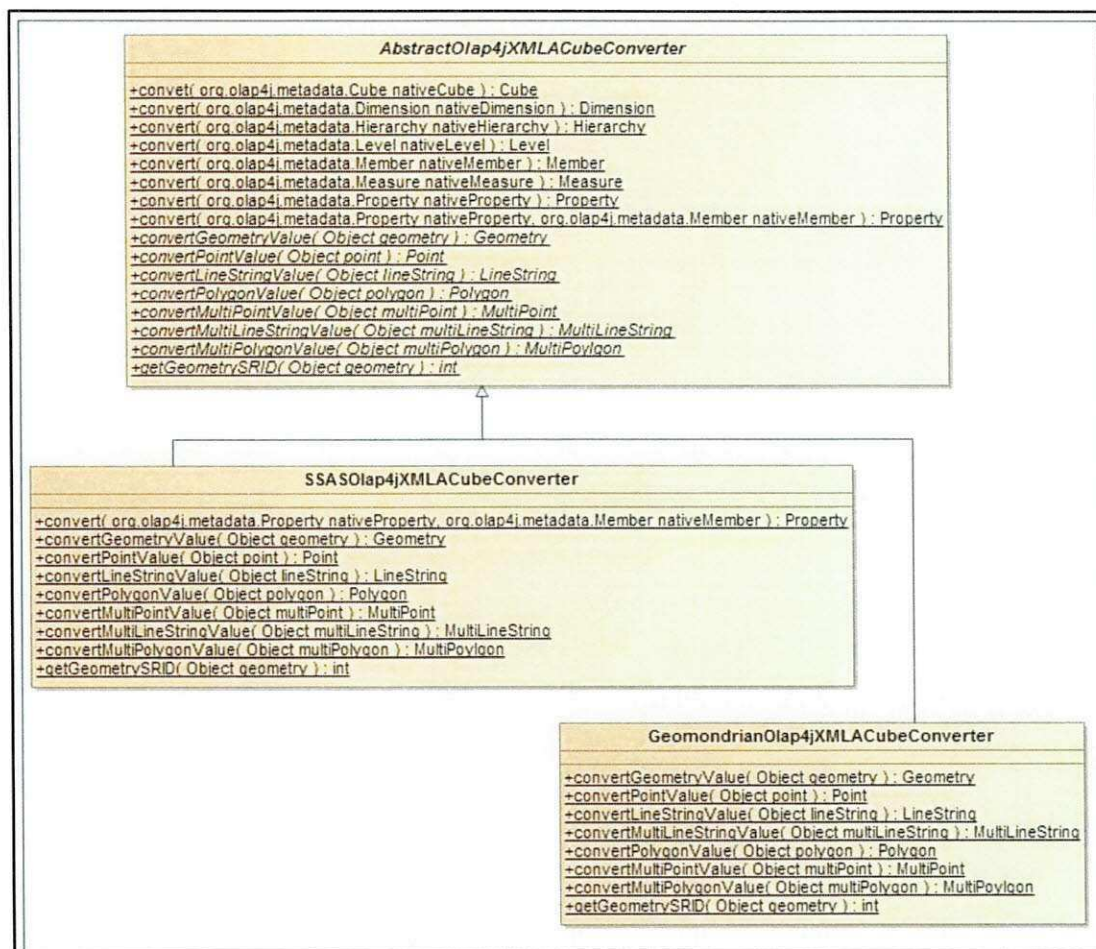
- Selezione uma fonte de dados:** A dropdown menu with "TCE SSAS" selected.
- Selezione um catalogo:** A dropdown menu with "CUBOS_SAGRES_GEO" selected.
- Selezione um cubo:** A list of cubes with "FAT_LIQUIDACAO" highlighted in blue. The list includes:
 - CUBO EMPENHO-PAGAMENTO-LIQUIDACAO
 - FAT_EMPENHO
 - FAT_LIQUIDACAO
 - FAT_PAGAMENTO
 - FAT_RECEITA_ORCAMENTARIA

Below the cube list is an "Alias: *" field with "LIQUIDACAO" entered. At the bottom is a "Conectar" button.

Após a seleção do cubo, seus metadados são carregados. Para isso, é preciso convertê-lo do formato nativo para o formato utilizado na solução. A carga dos metadados é realizada pela classe *Olap4jXMLACubeMetadataDAO* (Figura 57). A classe abstrata *AbstractOlap4jXMLACubeConverter* (Figura 56) implementa os métodos básicos necessários para a conversão dos cubos do formato nativo para o formato utilizado na solução e é utilizada pela classe *Olap4jXMLACubeMetadataDAO*.

Como XMLA não especifica um formato padrão para o transporte dos dados espaciais e o MDX [41], linguagem de consulta utilizada pelo XMLA, não especifica funções espaciais, a classe *AbstractOlap4jXMLACubeConverter* possui métodos abstratos referentes à parte espacial que depende da tecnologia utilizada pelo servidor XMLA. Os métodos abstratos devem ser implementados para garantir que os dados espaciais sejam devidamente convertidos em geometrias a serem utilizadas na solução. A API JTS [42] é utilizada para padronizar o formato das geometrias e está em conformidade com a especificação *Simple Feature Specification for SQL*, publicada e mantida pelo OGC.

Figura 56 – Diagrama de classes para a classe *AbstractOlap4jXMLACubeConverter*



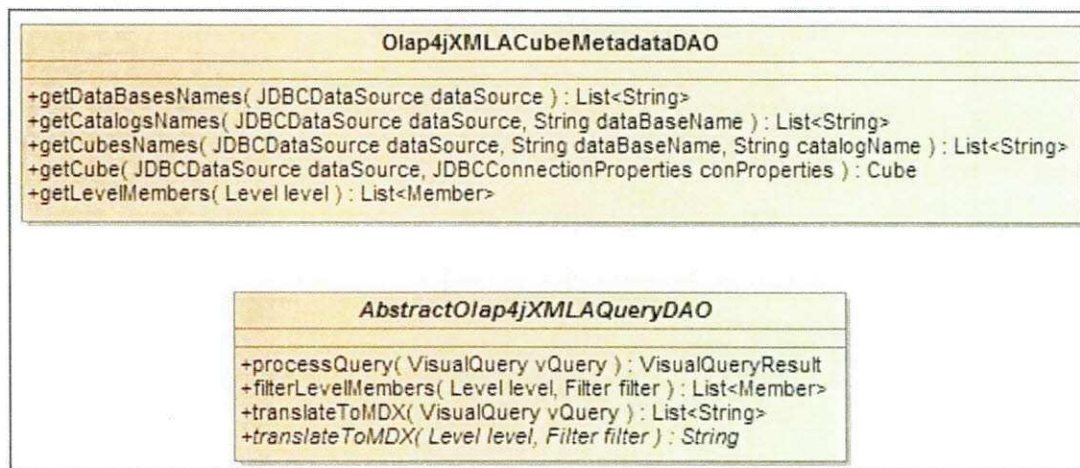
Para a carga dos dados, a classe *AbstractOlap4jXMLAQueryDAO* disponibiliza as funcionalidades básicas necessárias ao funcionamento da solução. A tradução da consulta visual para a linguagem MDX é responsabilidade dessa classe. O método *translateToMDX* da classe *AbstractOlap4jXMLAQueryDAO* recebe como parâmetro a consulta visual a ser traduzida para a linguagem MDX. O algoritmo responsável pela tradução da linguagem visual para a linguagem MDX é descrito pelo pseudo-código presente no Apêndice A.

Também é necessário implementar o método responsável por traduzir filtros em MDX (Figura 57). O método abstrato é necessário devido ao fato de a solução dispor de filtros que utilizam funções espaciais. E como essas funções não são padronizadas para o MDX, variam de acordo com a tecnologia utilizada. Ao executar os filtros, uma porção dos membros de um determinado nível hierárquico é selecionada. Os filtros são traduzidos para a linguagem MDX e executados no momento em que são criados, ou seja, durante a especificação visual da consulta.

Para garantir a sincronização OLAP/GIS para servidores que não dão suporte à análise SOLAP, os filtros espaciais são repassados para o módulo de geocodificação, onde eles são traduzidos para a linguagem SQL e executados, realizando uma seleção dos membros do nível hierárquico que será utilizada pela parte OLAP da análise.

Logo, para que o engenho XMLA acesse determinada tecnologia, é necessário implementar as classes abstratas descritas nesta seção.

Figura 57 – Diagrama de Classes para os DAOs do engenho XMLA



4.5. CONSIDERAÇÕES

Este capítulo apresentou uma proposta de *framework* para a análise SOLAP, por meio da definição de cubo espacial, da extensão espacial para a linguagem VizQL [24], da arquitetura e dos pontos de extensão do *framework*. O desenvolvimento do *framework* foi guiado por alguns requisitos, entre eles, a capacidade de acessar várias fontes de dados multidimensionais e dispor de operadores espaciais e de uma linguagem visual para as consultas.

Para contemplar o acesso a várias fontes de dados multidimensionais, foram criados, na arquitetura do *framework*, pontos de extensão que permitem que programadores o estendam a partir da implementação de algumas funcionalidades. O processo de extensão do *framework* foi descrito neste capítulo, bem como sua extensão para fontes que disponibilizam os dados multidimensionais via XMLA. Para dar suporte à análise SOLAP, a linguagem

VizQL foi estendida com uma nova divisória denominada “Camadas”, além de operadores espaciais topológicos e um novo tipo de painel para a exibição de mapas.

Assim, com a concepção do *framework* proposto nesta dissertação, cujo objetivo é de minimizar as lacunas identificadas nos trabalhos relacionados, realizou-se uma nova comparação avaliativa, com base na realizada na seção 3.9. A Tabela 3 exibe o novo quadro comparativo, onde uma célula contendo um X significa que a ferramenta apresenta a característica correspondente e a contendo um – (hífen) significa que a ferramenta não apresenta a característica correspondente.

Tabela 3 – Novo quadro comparativo

Soluções	Características									
	I	II	III	IV	V	VI	VII	VIII	IX	X
JMAP Spatial OLAP	X	X	X	X	-	-	-	X	X	X
GeoWOLAP	X	X	X	X	-	-	-	-	-	-
SOVAT	X	X	-	-	-	-	-	-	-	X
Piet	-	-	-	-	-	-	-	-	-	-
Golapware	-	X	X	-	-	-	X	X	-	X
Mapwarehouse	X	X	X	-	-	-	X	X	X	-
Polaris	X	X	X	X	-	-	-	-	-	X
DESOURB/Saiku	X	X	X	-	-	-	-	X	X	X
Framework proposto nesta dissertação	X	X	X	X	X	X	X	X	X	X

Como observado na Tabela 3, o *framework* proposto conseguiu alcançar as características propostas na seção 3.9, embora, por questões de escopo, apresente algumas limitações, a saber:

- **Operadores espaciais:** apenas os operadores topológicos foram disponibilizados no *framework*;
- **Restrições espaciais:** o *framework* não permite criar restrições por meio da seleção de uma geometria ou da definição de uma área no mapa;
- **Operadores OLAP:** o *framework* disponibiliza os operadores *roll-up*, *drill-down*, *slice and dice* e *pivot*, porém não dispõe de outros operadores OLAP, como *ranking*;
- **Operadores SOLAP:** o *framework* dispõe apenas dos operadores espaciais *spatial drill-down* e *roll-up*;

- **Visualização dos dados:** embora seja possível comparar os dados por meio de uma estrutura tabular, o *framework* não permite que eles sejam visualizados em diferentes *displays* que podem estar sincronizados.

Os pontos relevantes do *framework* proposto são: utilização de operadores espaciais topológicos; uma linguagem visual com suporte espacial para definir as consultas e as visualizações dos dados; e acesso várias fontes de dados multidimensionais. Os pontos críticos da implementação são: custo considerável na visualização de um grande número de mapas e na carga de grandes quantidades de dados, por exemplo, níveis com grande número de membros.

A seguir, será apresentado o exemplo prático realizado para validar o *framework* proposto.

5. AVALIAÇÃO E EXEMPLO PRÁTICO

Para avaliar a solução, o engenho SOLAP foi estendido para dar suporte às fontes multidimensionais *Microsoft SQL Server Analysis Services (SSAS)* e *GeoMondrian*. Também foi desenvolvido um exemplo prático destinado à análise das prestações de contas do Tribunal de Contas do Estado do Acre (TCE-AC).

5.1. EXTENSÕES DO ENGENHO SOLAP PARA SERVIDORES XMLA

O servidor de cubos SSAS faz parte da solução de BI da Microsoft e usa o XMLA como protocolo para se comunicar com aplicativos cliente. O *Analysis Services* não dá suporte a dados espaciais. O engenho SOLAP para XMLA foi estendido para integrar a solução proposta ao *Analysis Services*.

Ao se conectar com essa fonte de dados, o engenho SOLAP marca o *DataSource* utilizado como não espacial, o que obriga o *framework* a acessar os dados espaciais por meio do Módulo de Geocodificação. Antes da análise dos dados, o usuário deve geocodificar os membros do cubo para habilitar a análise espacial. Foram implementadas as classes necessárias para a extensão do engenho XMLA. O método abstrato *filterLevelMembers* da classe *AbstractOlap4jXMLAQueryDAO* (Figura 57), responsável por traduzir filtros em consultas MDX, foi implementado para permitir que o processamento de filtros espaciais seja atribuído ao Módulo de Geocodificação. Os dados espaciais originados da geocodificação são mantidos no SGBD PostgreSQL com extensão espacial PostGis. Também foi feito um mapeamento entre as funções espaciais utilizadas no *framework* e as disponíveis no PostGIS.

Os métodos abstratos da classe *AbstractOlap4jXMLACubeConverter* foram implementados pela classe concreta que herda a classe abstrata e são responsáveis por converter os objetos geométricos do PostGIS em objetos da API JTS.

O GeoMondrian é um servidor SOLAP código de fonte aberto, que provê a integração dos dados espaciais e cubos OLAP. Também pode ser interpretado como uma extensão espacial da linguagem MDX utilizada pelo Mondrian (*Pentaho Analysis Services*). O *DataSource* dessa fonte de dados é marcado como espacial pelo engenho SOLAP no momento da conexão, o que significa que os dados espaciais são recuperados diretamente dos cubos de dados e que é de responsabilidade do servidor SOLAP realizar as funções espaciais necessárias ao adequado funcionamento do *framework*. O MDX do GeoMondrian dispõe das

funções espaciais necessárias ao *framework*. A classe abstrata *AbstractOlap4jXMLAQueryDAO* é concretizada e dispõe de um método que é responsável por traduzir filtros espaciais em consultas MDX. Os dados espaciais são enviados via servidor XMLA, no formato WKT especificado pelo OGC. A classe concreta que herda a classe abstrata *AbstractOlap4jXMLACubeConverter* tem os métodos necessários para converter as geometrias do formato WKT para objetos da API JTS.

5.2. EXEMPLO PRÁTICO APLICADO AO TRIBUNAL DE CONTAS DO ESTADO DO ACRE

Esta seção apresenta o exemplo prático de uma aplicação SOLAP referente à prestação de contas do Tribunal de Contas do Estado do Acre. Nessa aplicação, informações sobre empenhos, liquidações e pagamentos serão analisados, visando auxiliar o processo de tomada de decisão relacionado à obtenção de estratégias de gestão mais eficazes e à fiscalização das contas públicas.

A fim de elucidar os conceitos envolvidos no exemplo, um documento, obtido no site do Tribunal de Contas da União (TCU) [43] contendo os conceitos básicos, está disponível no Anexo A.

5.2.1. Esquema conceitual

A aplicação para o exemplo prático proposto analisa cubos disponibilizados pelas ferramentas Microsoft SSAS e GeoMondrian, via servidor XMLA. Como o servidor de cubos da Microsoft não dá suporte a dados espaciais, os membros dos níveis foram geocodificados. Assim, medidas espaciais não foram utilizadas nesse exemplo.

Três cubos estão disponíveis para a análise: Empenho, Liquidação e Pagamento. O cubo Empenho é disponibilizado pelo servidor GeoMondrian, enquanto que os demais são disponibilizados pelo SSAS. Os esquemas conceituais apresentados nas Figura 58, Figura 59 e Figura 60 modelam o DW Espacial utilizado para a implementação dos cubos. Os fatos a serem analisados são representados pelas medidas valor do empenho, valor da liquidação e valor do pagamento. Além das medidas modeladas no DW Espacial, os cubos contam com mais duas medidas calculadas: quantidade e média dos valores.

Os fatos apresentam as seguintes dimensões em comum: *Ação, Fornecedor, Função, Natureza da Despesa, Programa, Subelemento de Despesa, Subfunção, Tempo, Tipo da Fonte*

de *Recurso Orçamentária* e *Unidade Orçamentária*. A dimensão *Modalidade da Despesa* é comum aos cubos *Empenho* e *Pagamento*. A dimensão *Empenho Resumo* é comum aos cubos *Liquidação* e *Pagamento*, a dimensão *Tipo de Empenho* caracteriza apenas as medidas do cubo *Empenho*, enquanto que a dimensão *Conta Bancária* caracteriza somente as medidas do cubo *Pagamento*.

Algumas dimensões tiveram seus membros organizados em hierarquias: *Ação*, *Fornecedor*, *Subelemento*, *Conta Bancária*, *Tempo* e *Unidade Orçamentária*. Os membros das dimensões foram organizados em níveis de detalhes da seguinte forma:

- *Dimensão Ação*: membros organizados em tipo de ação e ação;
- *Dimensão Fornecedor*: membros organizados em tipo de fornecedor, cpf cnpj e nome do fornecedor;
- *Dimensão Subelemento*: membros organizados em elemento de despesa e subelemento;
- *Dimensão Conta Bancária*: membros organizados em banco, agência, tipo de conta e conta;
- *Dimensão Tempo*: membros organizados em ano, mês e dia;
- *Dimensão Unidade Orçamentária*: membros organizados em estado, mesorregião, microrregião, município, unidade gestora e unidade orçamentária.

A dimensão *Unidade Orçamentária* é uma dimensão espacial, pois dispõe de atributos e hierarquia espaciais: *Nome Estado*, *Nome Mesorregião*, *Nome Microrregião*, *Nome Município* e *Hierarquia Estado – Meso – Micro – Município*. A porção do DW Espacial referente ao cubo *Empenho* foi migrada para o SGBD PostgreSQL para que o GeoMondrian tivesse acesso a esses dados.

Figura 58 – Modelo do DW utilizado para construir o cubo Empenho

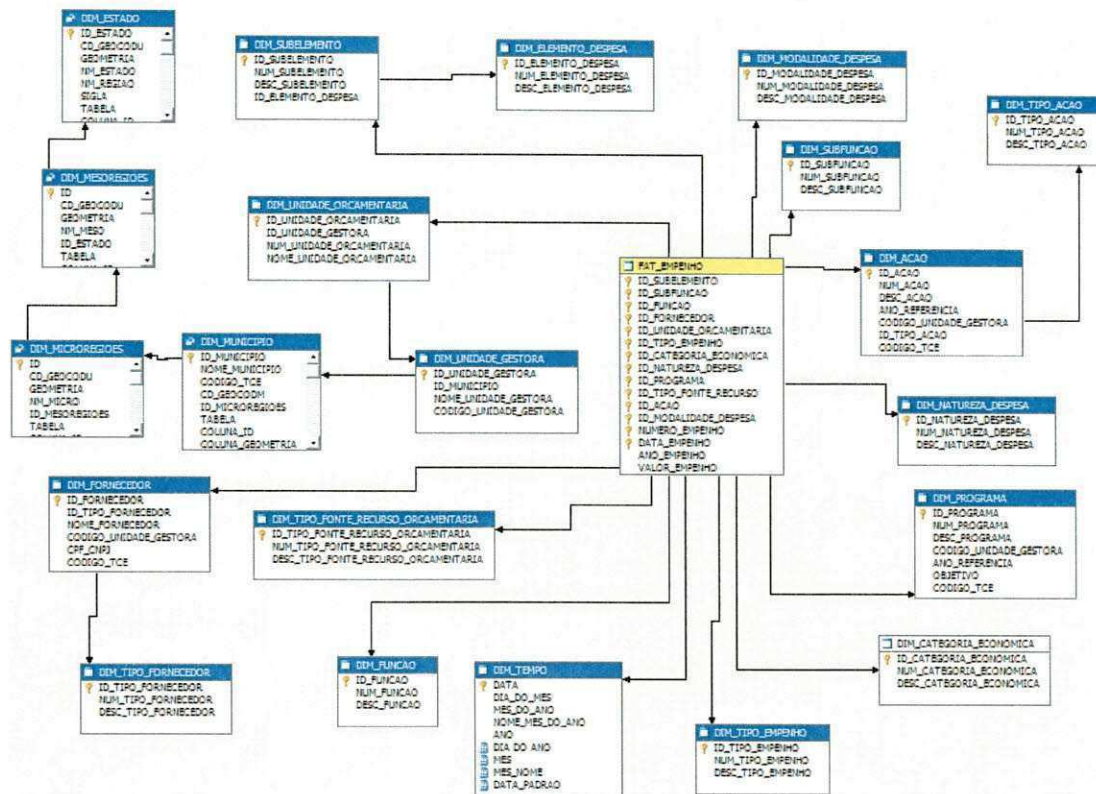


Figura 59 – Modelo do DW utilizado para o cubo Liquidação

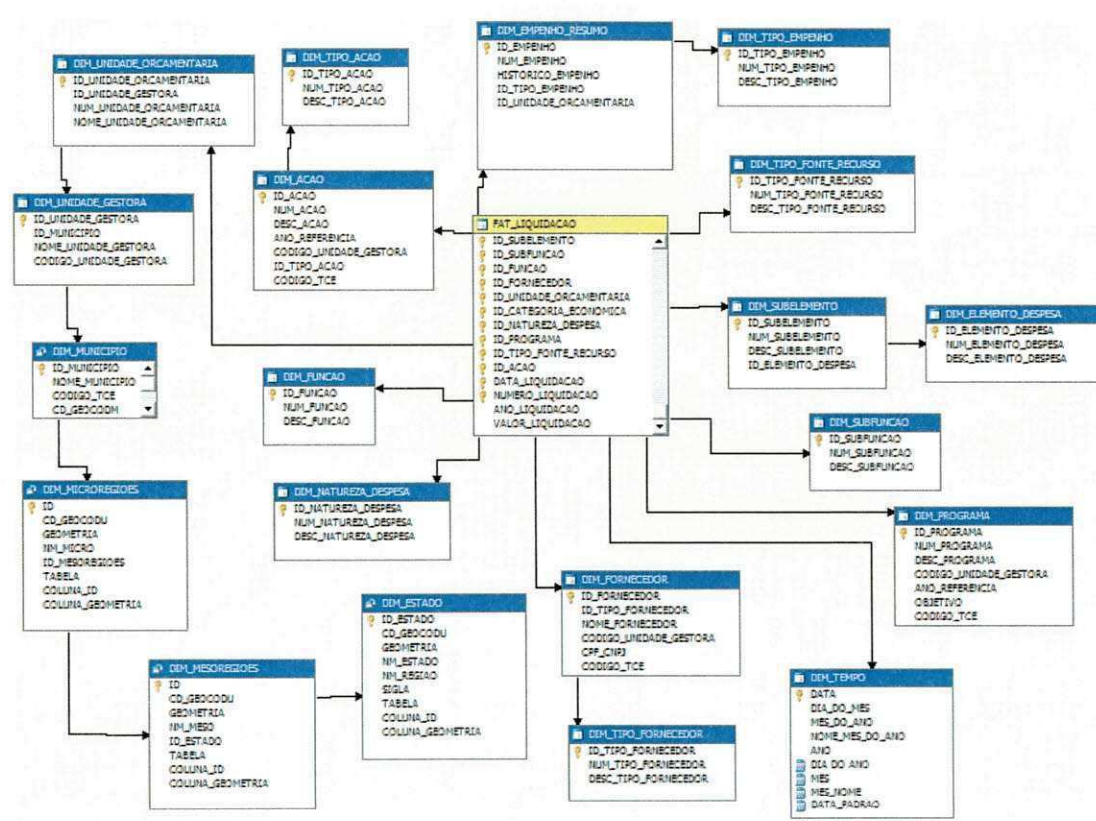
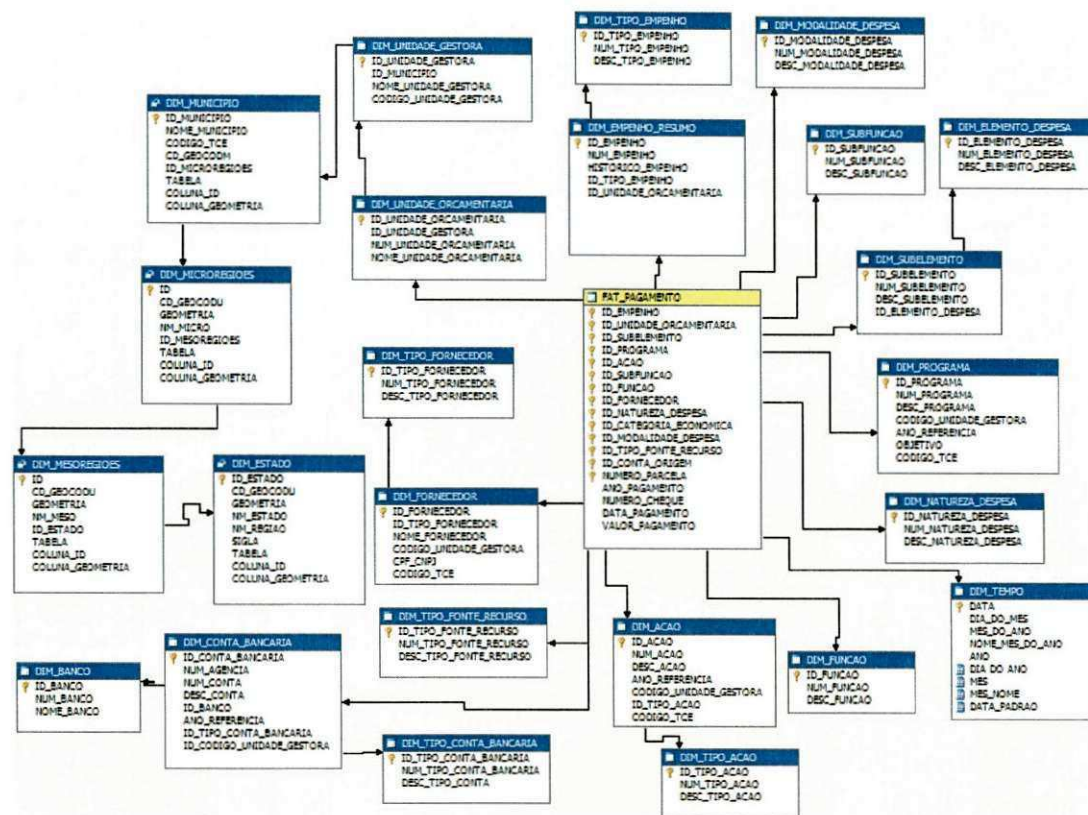


Figura 60 – Modelo do DW para o cubo Pagamento



5.2.2. Consultas

As consultas a seguir apresentam as funcionalidades relacionadas à análise espacial em fonte de dados multidimensionais.

Consulta 1

“Exiba no mapa a média dos valores de empenhos e detalhe por estado, mesorregião, microrregião e município.”

Para realizar essa consulta, o cubo *Empenho* foi utilizado. A medida *Média Valor Empenho* foi adicionada à divisória “Colunas”, e a hierarquia *Hierarquia Estado – Meso – Micro – Município* da dimensão *Dim Unidade Orçamentária* foi adicionada à divisória “Camadas”. Essa hierarquia é composta de quatro níveis espaciais: *Estado, Mesorregião, Microrregião e Município*.

Ao adicionar a hierarquia à divisória, o nível menos detalhado da hierarquia é exibido junto com o ícone de navegação + que é utilizado para aumentar o nível de detalhes (Figura 61). Para detalhar os dados, o ícone + foi utilizado para os níveis *Estado*, *Mesorregião* e *Microrregião* respectivamente (Figura 62, Figura 63 e Figura 64). A visualização dos dados foi especificada selecionando-se o tipo de painel espacial e a marcação do tipo texto, onde a medida selecionada foi utilizada como rótulo para as geometrias. Se a geometria não tiver um valor de medida associado, ela não será exibida no mapa. O objetivo dessa consulta é de exemplificar a navegação nas hierarquias e a visualização do tipo de texto em painéis espaciais (mapas).

Figura 61 – Resultado da consulta 1 - nível de detalhe para o estado do Acre

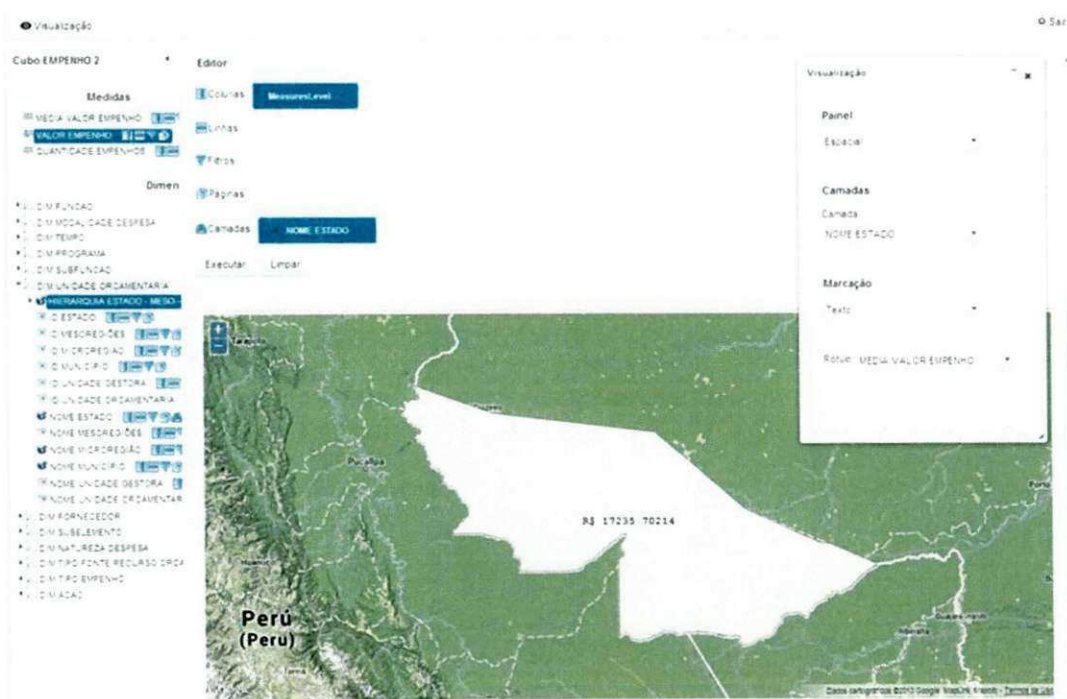


Figura 62 – Resultado da Consulta 1 - nível de detalhe para as mesorregiões do Acre

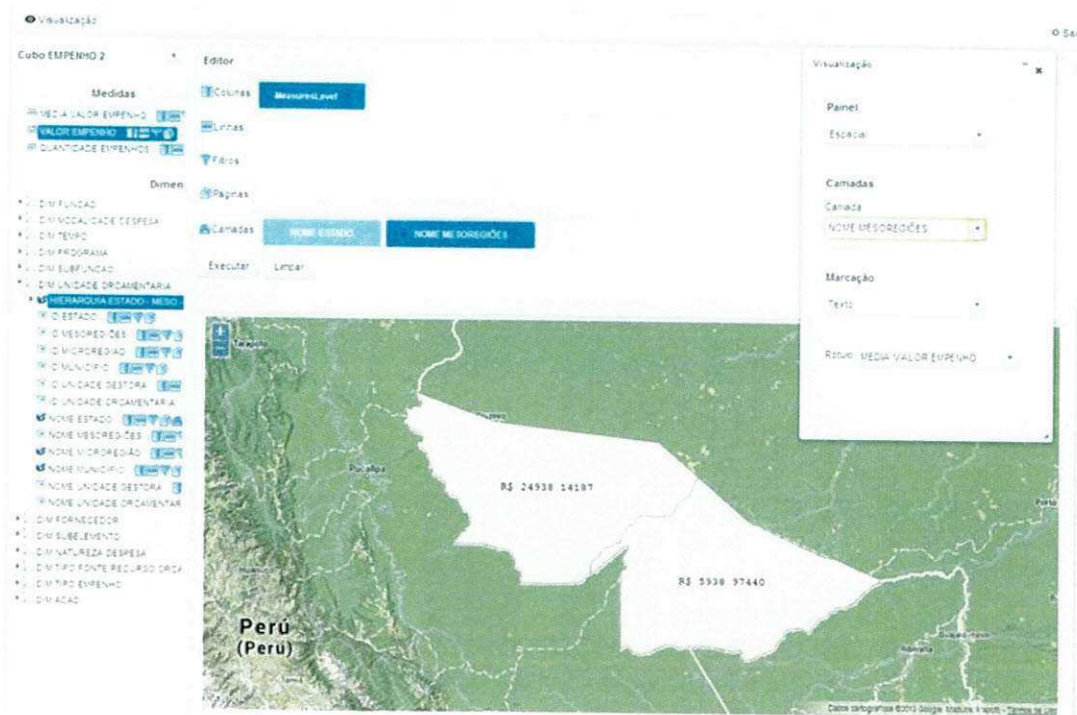


Figura 63 – Resultado da Consulta 1 - nível de detalhe para as microrregiões do Acre

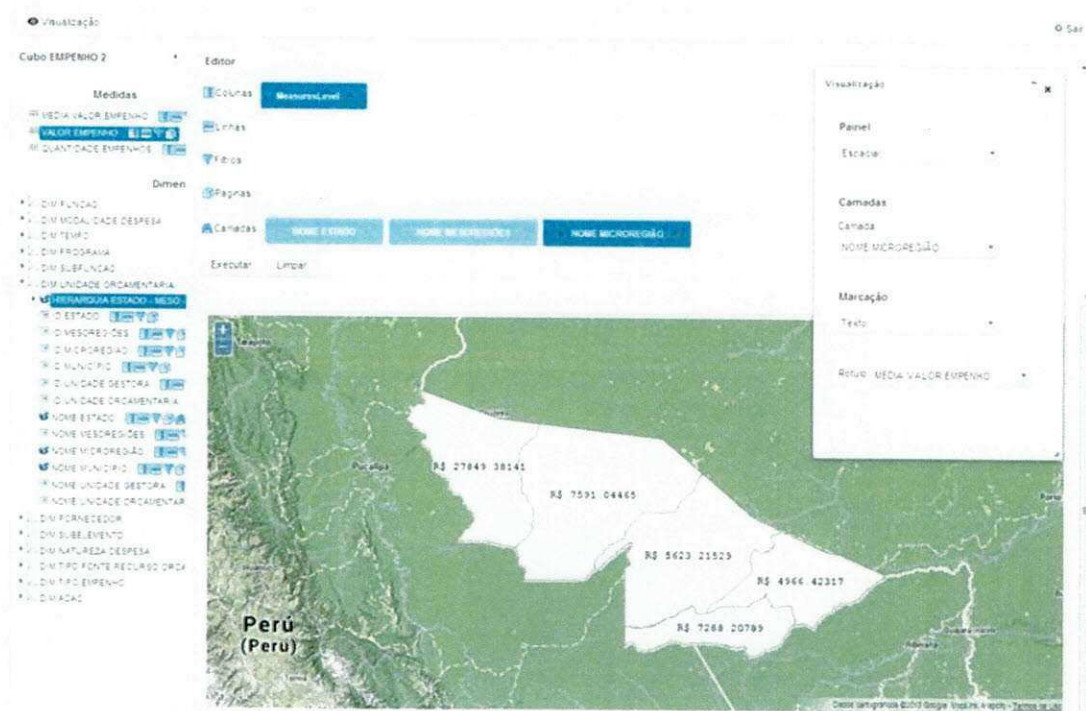
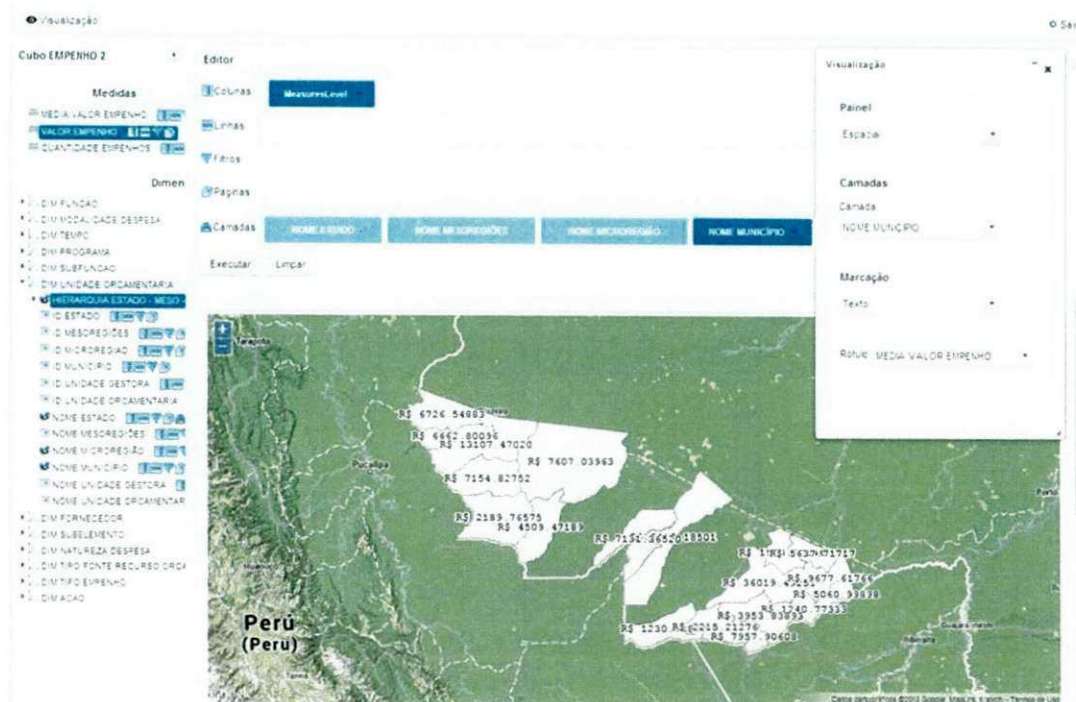


Figura 64 – Resultado da consulta 1 - nível de detalhe para os municípios do Acre



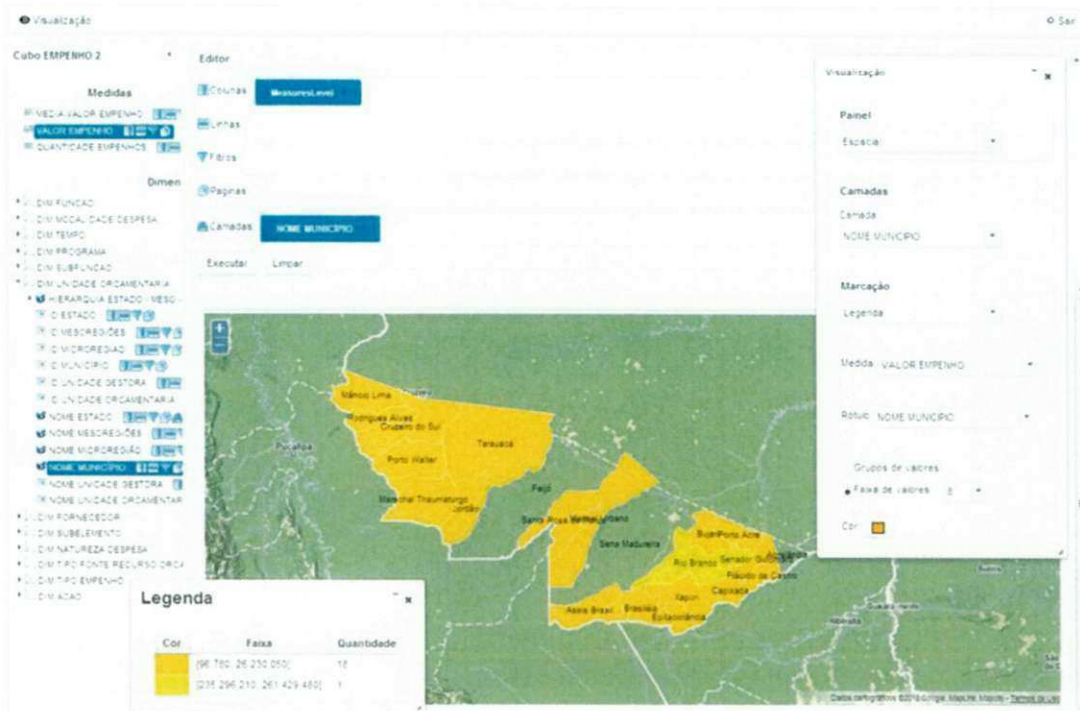
Consulta 2

“Exiba um mapa temático com a soma dos valores de empenhos para cada município.”

Para essa consulta, ainda utilizando o cubo *Empenho*, a medida *Valor Empenho* foi adicionada à divisória “Colunas” enquanto a hierarquia espacial *Nome Município* foi adicionada à divisória “Camadas”. O tipo de marcação *Legenda* foi utilizado para visualizar os dados no mapa (Figura 65).

A legenda foi criada para a medida *Valor Empenho*, é do tipo faixa de valores, e a amplitude dos valores das medidas é utilizada para agrupar as geometrias. Também é possível criar legendas do tipo grupo, em que os registros são divididos uniformemente nas faixas de valores. O nível *Nome Município* foi utilizado como rótulo para as geometrias. Essa consulta foi utilizada para exemplificar a visualização do tipo *Legenda* em mapas.

Figura 65 – Resultado da consulta 2



Consulta 3

“Qual a soma dos valores liquidados nos municípios vizinhos a Rio Branco para as funções Administração, Agricultura e Legislativa?”

Essa consulta demonstra a utilização dos filtros espaciais disponíveis no *framework*. Nesse exemplo, o cubo *Liquidação* foi utilizado, a medida *Valor Liquidação* foi adicionada à divisória “Colunas”, a hierarquia *Descrição Função*, à divisória “Linhas”, e a hierarquia *Nome Município*, à divisória “Camadas”.

Os membros do nível *Descrição Função* foram filtrados (Figura 66), um filtro geográfico foi aplicado aos municípios, e o operador espacial *Touche*s foi utilizado para filtrar os vizinhos a Rio Branco (Figura 67). O resultado do filtro geográfico pode ser observado na aba do filtro convencional (Figura 68). Para visualizar os dados, foi utilizada a marcação *Legenda* em painéis espaciais. A Figura 69 apresenta o resultado da consulta.

Figura 66 – Filtro convencional para as funções

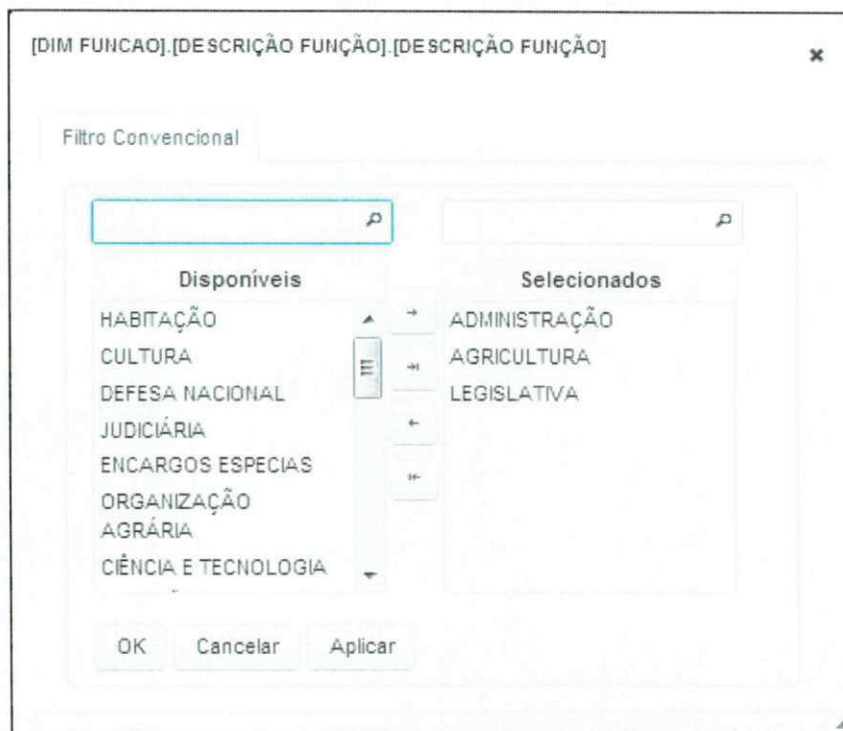


Figura 67 – Filtro espacial para os municípios

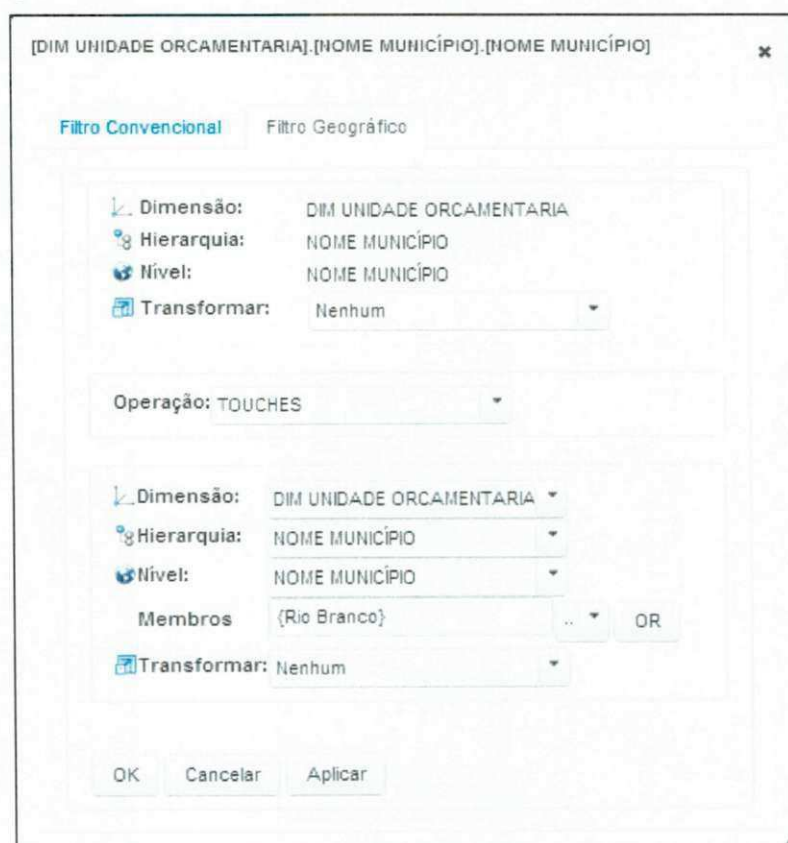


Figura 68 – Resultado do filtro espacial refletido no filtro convencional

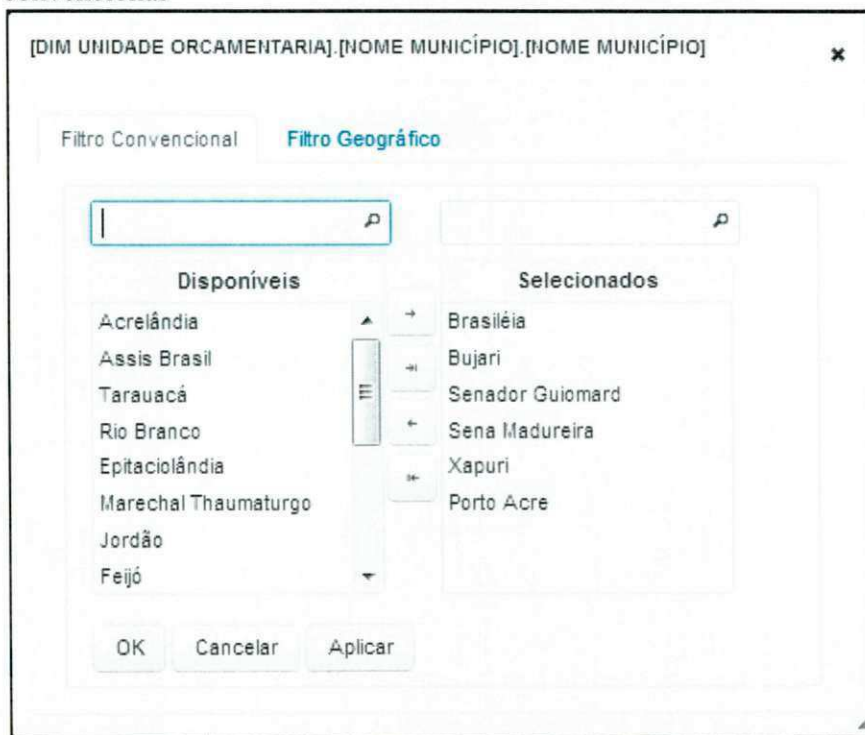
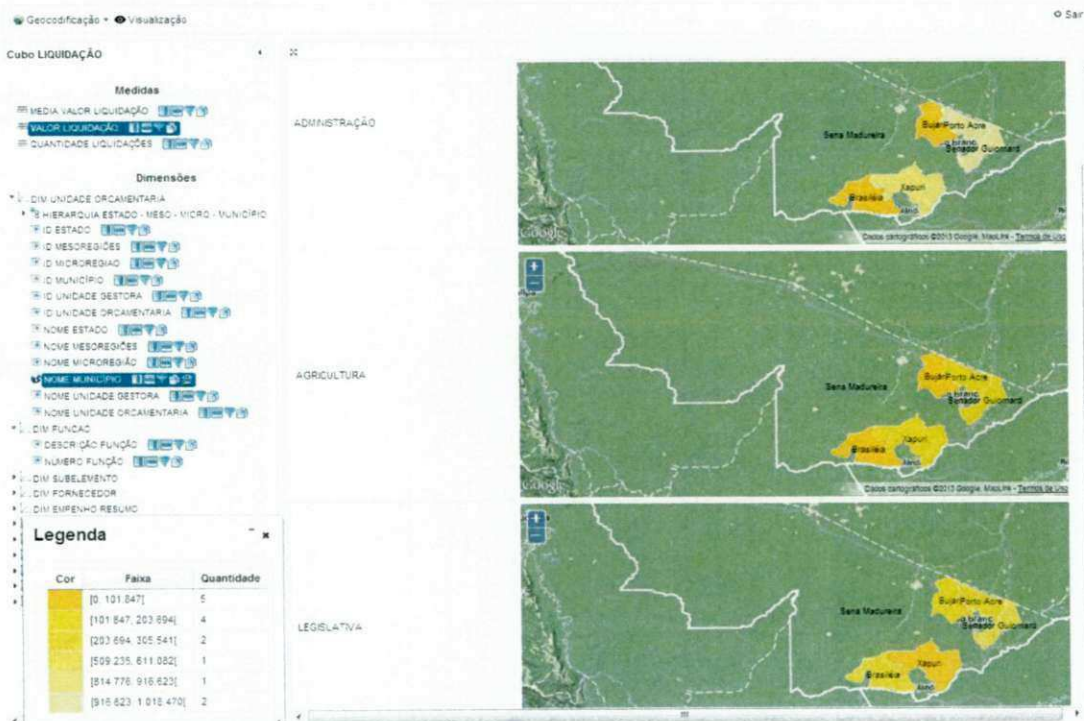


Figura 69 – Resultado da consulta 3



Consulta 4

“Exiba gráficos de barras contendo a soma dos valores pagos por natureza da despesa para o tipo de empenho Global nos municípios contidos nas microrregiões Sena Madureira ou Tarauacá.”

No cubo *Pagamento*, para recuperar apenas os valores associados ao tipo de empenho Global, adicionou-se a hierarquia *Descrição Tipo Empenho* na divisória “Filtros” onde o tipo *Global* foi selecionado. A medida *Valor Pagamento* e a hierarquia *Nome Município* foram adicionadas à divisória “Linhas”. A hierarquia *Descrição Natureza Despesa* foi adicionada à divisória “Colunas”. Para selecionar os municípios contidos nas microrregiões Sena Madureira ou Tarauacá, utilizou-se um filtro geográfico com a função espacial *Within* e a função espacial *Buffer* aplicada às microrregiões para resolver problemas relacionados às fronteiras (Figura 70). O tipo de painel utilizado nessa consulta foi o convencional, com a marcação Gráfico do tipo Barras (Figura 71).

Figura 70 – Filtro para os municípios contidos nas microrregiões Sena Madureira ou Tarauacá

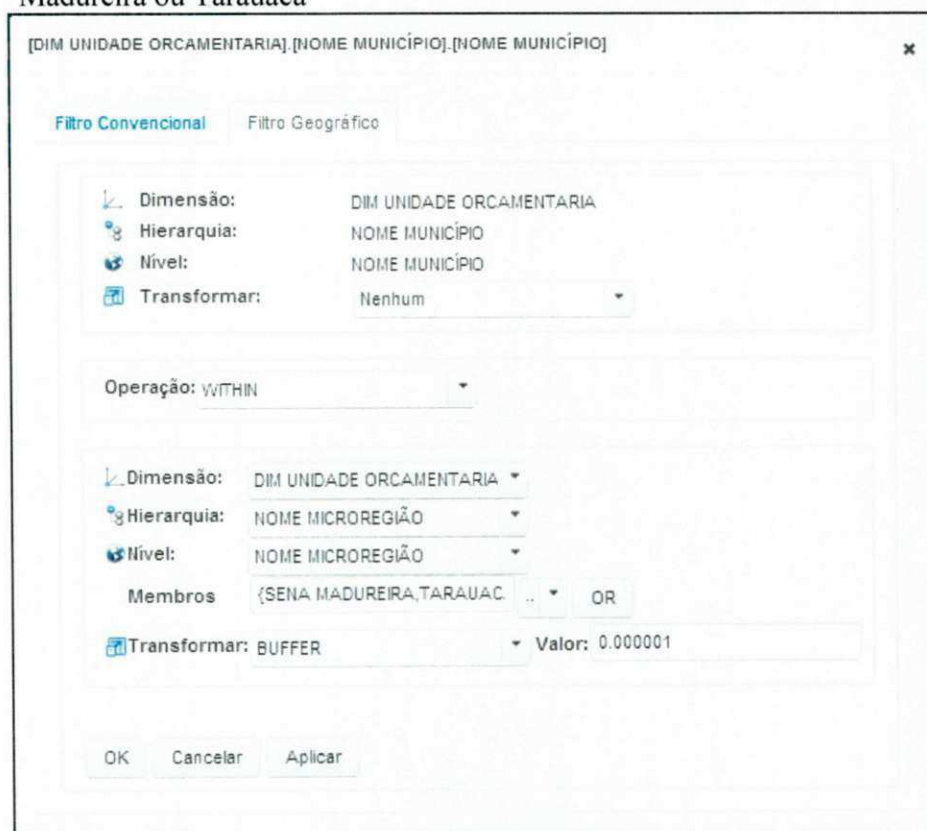
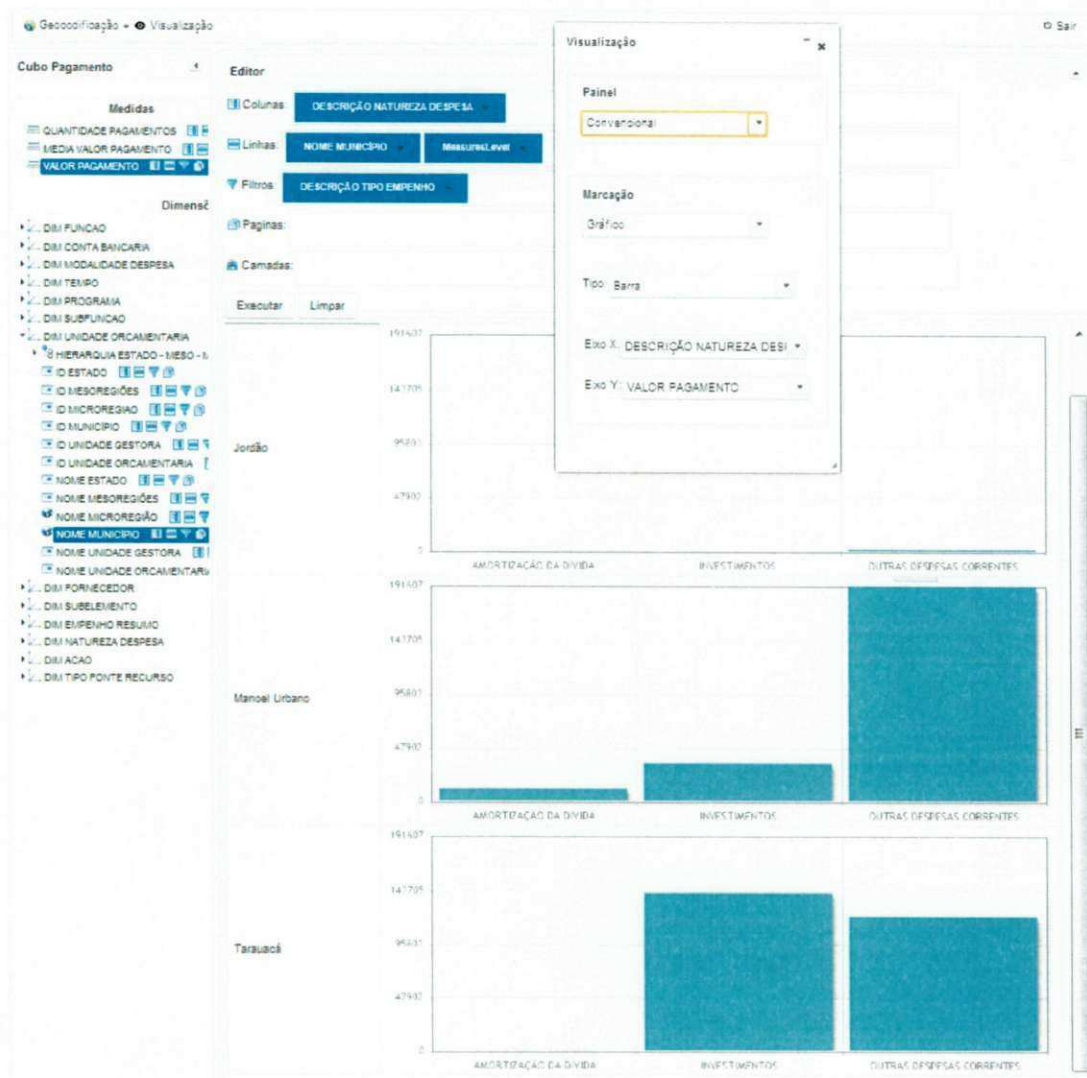


Figura 71 – Resultado da consulta 4



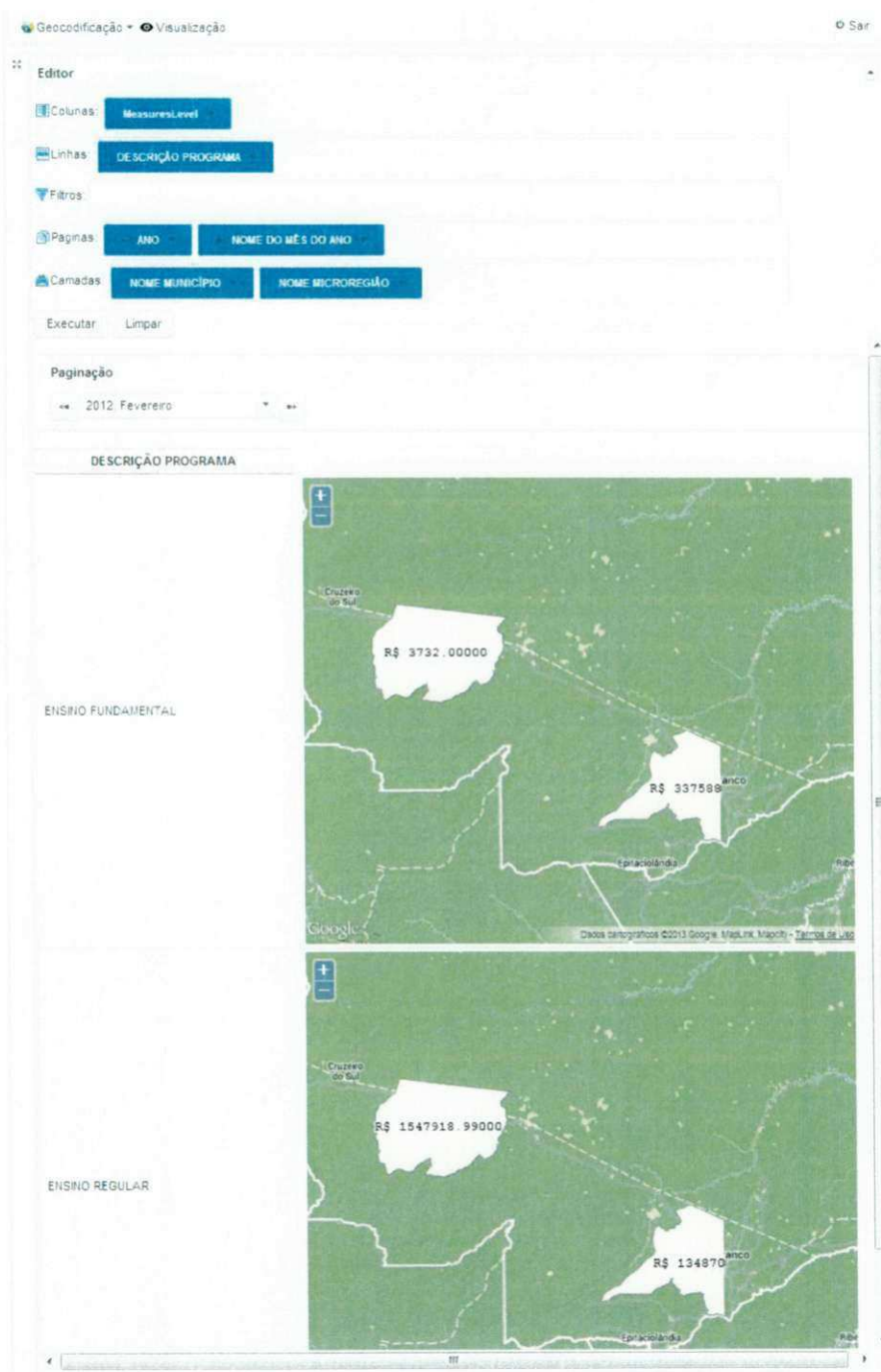
Consulta 5

“Quanto foi gasto com o ensino de graduação, o ensino fundamental e o ensino regular no ano de 2012, detalhado por mês, na microrregião Rio Branco e no município Tarauacá?”

Para exemplificar a paginação e a sobreposição de camadas, a medida *Valor Pagamento* foi adicionada à divisória “Colunas”, a hierarquia *Descrição Programa*, à divisória “Linhas”, a hierarquia *Ano – Mês – Dia*, à divisória “Páginas”, e as hierarquias *Município* e *Microrregião*, à divisória “Camadas”. O membro 2012 do nível *Ano* foi selecionado e realizou-se um *drill-down* para detalhar as páginas por mês. O membro *Rio Branco* do nível *Microrregião* e o membro *Tarauacá* do nível *Município* foram selecionados.

A visualização foi realizada no painel espacial com a marcação Texto. O resultado pode ser observado na Figura 72.

Figura 72 – Resultado da consulta 5



5.3. CONSIDERAÇÕES

Este exemplo prático teve como objetivo aplicar as funcionalidades disponíveis para a análise SOLAP propostas nesta dissertação. A utilização do *framework* trouxe resultados satisfatórios, mostrando a experiência do usuário ao realizar consultas SOLAP e ao visualizar os resultados das consultas em um conjunto de dados provenientes dos processos de prestação de contas do TCE-AC.

6. CONCLUSÕES E TRABALHOS FUTUROS

6.1. CONCLUSÕES

Várias soluções visando à integração dos dados espaciais e dos dados multidimensionais foram propostas nos últimos anos, conforme apresentado nesta dissertação. O principal objetivo dessas soluções era propor melhorias no processo analítico dos dados, proporcionando um ambiente único para a análise dos componentes espaciais das fontes multidimensionais.

A partir do levantamento do estado da arte, apresentado no Capítulo 3, foi possível relacionar os pontos fortes e os fracos das principais soluções existentes e constatar que, por não haver um padrão para disponibilizar os cubos espaciais na Web, essas soluções não disponibilizam ou apresentam técnicas para analisar cubos espaciais oriundos de outras fontes. Com base nas questões observadas, originou-se uma lista de requisitos necessários a uma ferramenta de análise SOLAP, cujos principais requisitos são: promover o acesso a várias fontes de dados multidimensionais e a geocodificação dos dados; dar suporte aos operadores espaciais topológicos e dispor de uma linguagem visual para a especificação de consultas e para a visualização dos dados.

Diante das questões levantadas e com o intuito de contemplar os requisitos mencionados nesta dissertação, foi apresentada uma proposta de *framework* capaz de realizar a análise espacial dos dados provenientes de várias fontes de dados espaciais multidimensionais, tendo como principal esforço a extensão do engenho SOLAP para o acesso às fontes de dados. O *framework* dispõe de um conjunto de interfaces gráficas que contribuem para que o usuário possa criar conexões para o acesso aos cubos de dados, realize análises mediante a especificação visual de consultas e visualizações dos dados, crie filtros convencionais ou espaciais para restringir os dados, geocodifique membros das dimensões, compare os dados por meio de uma estrutura tabular e utilize mapas para visualizar os dados.

No *framework* proposto nesta dissertação, o modelo de cubo espacial definido na seção 4.1 mostrou-se satisfatório, ao permitir a análise espacial em cubos provenientes de variadas fontes. A extensão da linguagem de especificação visual VizQL [24] abriu o leque de possibilidades de análise espacial, uma vez que possibilitou o uso de operadores espaciais topológicos e a exibição dos dados em mapas, com sobreposição de camadas. O *framework* também deu suporte à análise espacial, seguiu a abordagem federada de integração dos dados

e disponibilizou um serviço de geocodificação que possibilita a análise espacial em servidores puramente OLAP.

Para que o usuário possa acessar várias fontes de dados, pontos de extensões foram criados e documentados para que terceiros possam estendê-lo. Essa característica classifica a solução como um *framework*. Os servidores de cubos SSAS e GeoMondrian foram acessados por meio de extensões do *framework* proposto, o que permitiu a realização de avaliações concretas de funcionalidade por meio de um exemplo prático baseado em situações e problemas reais do cotidiano, o que mostrou que é possível realizar, com eficiência e praticidade, análises espaciais em cubos provenientes de diferentes fontes, obtendo informações relevantes para solucionar os problemas reais relacionados ao processo de prestação de contas do TCE-AC.

A realização de uma avaliação comparativa entre as soluções relacionadas e o *framework* proposto nesta dissertação mostrou que esse *framework* conseguiu endereçar as características necessárias a uma solução SOLAP, como pode ser observado na Tabela 3. Assim, conclui-se que este trabalho de dissertação alcançou os seus objetivos, tendo como principal contribuição uma arquitetura com a qual se analisam cubos espaciais provenientes de diversas fontes multidimensionais e que atendem, de forma satisfatória, aos requisitos propostos.

Também se conclui que vários pontos de expansão podem ser explorados em trabalhos futuros, o que contribuirá para a construção de um *framework* mais robusto. A seguir, são detalhados alguns pontos de expansão.

6.2. TRABALHOS FUTUROS

Nesta seção, são abordados alguns pontos que foram observados no decorrer deste trabalho e que podem ser explorados.

Interoperabilidade: Apesar de essa solução possibilitar a análise de cubos de dados provenientes de diversas fontes, ela não permite a interoperabilidade entre cubos. Sugere-se, como trabalho futuro, o desenvolvimento de um módulo para a interoperabilidade entre cubos que faça uso do modelo formal proposto por Salehi et al. [44].

Operadores SOLAP: Ruiz e Times [45] definem e classificam um conjunto de operadores SOLAP. Visando dar um suporte melhor para a análise SOLAP, tem-se como trabalho futuro a adição de novos operadores para essa solução.

Disponibilização dos dados: A adaptação da arquitetura do *framework* proposto para disponibilizar serviços via *Web Services* é outro trabalho futuro. O OGC [10] reuniu esforços para padronizar serviços de GeoBI, porém ainda não se tem resultado.

Usabilidade: A usabilidade das interfaces propostas neste trabalho não foi avaliada, razão pela qual se sugere um estudo avaliativo da interface e o possível melhoramento, por meio da adição de funcionalidades como *drag-and-drop*.

Visualização dos dados: A necessidade de novas formas de visualização dos dados pode ser sugerida. Adição de novas marcações e o suporte para mais propriedades de retina [24] são exemplos dessa atividade.

Visualização automática: Os usuários dessa solução podem não dominar conceitos de estatística e/ou não saber qual a melhor forma de dispor os dados para obterem informações mais precisas. Por isso, outro trabalho futuro seria a implementação de um módulo de sugestão para a visualização dos dados.

Representação matricial para dados espaciais: Dados espaciais, denominados de matriciais ou *raster*, não são suportados por essa solução. A adição do suporte, bem como técnicas de análise para esses dados são uma forma de contribuição.

Mineração de dados: Técnicas de mineração de dados podem ser propostas para aperfeiçoar a análise e o processo de tomada de decisão.

REFERÊNCIAS

- [1] S. Chaudhuri, U. Dayal e V. Narasayya, “An overview of business intelligence technology,” *Commun. ACM*, vol. 54, n. 8, pp. 88-98, aug 2011.
- [2] C. Franklin, “An introduction to geographic information systems: linking maps to databases,” *Database*, vol. 15, n. 2, pp. 12-21, apr 1992.
- [3] F. M. Cardoso, *Um framework para data warehouse espacial [dissertação]*, Campina Grande: Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2009.
- [4] S. Rivest, Y. Bédard, M.-J. Proulx, M. Nadeau, F. Hubert e J. Pastor, “SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, n. 1, pp. 17-33, 2005.
- [5] J. Silva, V. Times, R. Fidalgo e R. Barros, “Providing Geographic-Multidimensional Decision Support over the Web,” em *Web Technologies Research and Development - APWeb 2005*, vol. 3399, Y. Zhang, K. Tanaka, J. Yu, S. Wang e M. Li, Eds., Springer Berlin Heidelberg, 2005, pp. 477-488.
- [6] M. Scotch e B. Parmanto, “SOVAT: Spatial OLAP Visualization and Analysis Tool,” em *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 6 - Volume 06*, Washington, DC, USA, 2005.
- [7] A. Escribano, L. Gomez, B. Kuijpers e A. A. Vaisman, “Piet: a GIS-OLAP implementation,” em *Proceedings of the ACM tenth international workshop on Data warehousing and OLAP*, New York, NY, USA, 2007.
- [8] J. Silva, *GeoMDQL: Uma linguagem de consulta geográfica e multidimensional [tese]*, Recife, PE: Centro de Informática, Universidade Federal de Pernambuco, 2008.
- [9] E. Dubé, T. Badard e Y. Bédard, “XML encoding and Web Services for Spatial OLAP data cube exchange: an SOA approach,” *CIT*, vol. 17, n. 4, pp. 347-358, 2009.
- [10] Open Geospatial Consortium (OGC), “Open Geospatial Consortium | OGC(R),” [Online]. Available: <http://www.opengeospatial.org/>. [Acesso em 18 Julho 2013].
- [11] OpenGIS®, OGC 09-044r2 , “Geospatial Business Intelligence (GeoBI) [White Paper],” Março 2011. [Online]. Available: <http://www.opengeospatial.org/pressroom/papers..>
- [12] S. Shekhar e S. Chawla, *Spatial Databases: A Tour*, Prentice Hall, 2003.
- [13] C. Date, *An Introduction to Database Systems*, 8 ed., Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.
- [14] H. J. Watson e B. H. Wixom, “The Current State of Business Intelligence,” *Computer*, vol. 40, n.

- 9, pp. 96-99, sep 2007.
- [15] W. Inmon, BUILDING THE DATA WAREHOUSE (4th Ed.), Wiley India Pvt. Limited, 2005.
- [16] C. S. Jensen, T. B. Pedersen e C. Thomsen, Multidimensional Databases and Data Warehousing, Morgan & Claypool Publishers, 2010.
- [17] P. Y. Caron, "Etude du potentiel de OLAP pour supporter l'analyse spatio-temporelle. Mémoire de M. Sc., Département des sciences géomatiques, Faculté de foresterie et géomatique, Université Laval," *Université Laval*, n. 132, 1998.
- [18] S. Rivest, Y. Bédard e Marchand, "Towards better support for spatial decision-making: Defining the characteristics of Spatial On-Line Analytical Processing (SOLAP)," em *Geomatica, the journal of the Canadian Institute of Geomatics* 2001.
- [19] Y. Bédard, T. Merrett e J. Han, "Fundamentals of spatial data warehousing for geographic knowledge discovery," *Geographic data mining and knowledge discovery*, vol. 2, pp. 53-73, 2001.
- [20] J. Han, N. Stefanovic e K. Koperski, "Selective materialization: An efficient method for spatial data cube construction," em *Research and Development in Knowledge Discovery and Data Mining*, vol. 1394, X. Wu, R. Kotagiri e K. Korb, Eds., Springer Berlin Heidelberg, 1998, pp. 144-158.
- [21] Y. B. M. J. P. M. N. S. Rivest, "SOLAP: a new type of user interface to support spatio-temporal multidimensional data exploration and analysis," em *Proceedings of the ISPRS Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis, October 2-3, Quebec, Canada, 2003*.
- [22] S. Bimonte, A. Tchounikine e M. Miquel, "Towards a spatial multidimensional model," em *Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, New York, NY, USA, 2005.
- [23] M. Alexander e B. Jelen, em *Pivot Table Data Crunching*, Que Publishing, 2006, p. 288.
- [24] C. Stolte e P. Hanrahan, "Polaris: A System for Query, Analysis and Visualization of Multi-Dimensional Relational Databases," em *Proceedings of the IEEE Symposium on Information Visualization 2000*, Washington, DC, USA, 2000.
- [25] J. Bertin, "Readings in information visualization," S. K. Card, J. D. Mackinlay e B. Shneiderman, Eds., San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1999, pp. 62-65.
- [26] W. S. Cleveland, The elements of graphing data, Belmont, CA, USA: Wadsworth Publ. Co., 1985.
- [27] E. R. Tufte, The visual display of quantitative information, Cheshire, CT, USA: Graphics Press, 1986.

- [28] “Mondrian,” [Online]. Available: <http://mondrian.pentaho.com/>. [Acesso em 04 Junho 2013].
- [29] R. Fidalgo, *Uma Infra-Estrutura para Integração de Modelos, Esquemas e Serviços Multidimensionais e Geográficos [Tese de Doutorado]*, Centro de Informática - Universidade Federal de Pernambuco, 2005.
- [30] R. L. Fonseca, R. N. Fidalgo, J. Silva e V. C. Times, “Geodwcase: Uma ferramenta para projeto de data warehouses geográficos,” em *In Brazilian Symposium on Databases(SBBD), Demos Session, 2007*.
- [31] F. S. M. B. J. V. A. Lamas, “Creación de un módulo espacial OLAP para,” *VII JORNADAS DE SIG LIBRE, 2013*.
- [32] ORACLE, “JavaServer Faces Technology,” [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>. [Acesso em 15 Junho 2013].
- [33] PrimeFaces, “PrimeFaces,” [Online]. Available: <http://primefaces.org/>. [Acesso em 15 Junho 2013].
- [34] ORACLE, “Oracle and Java | Technologies | Oracle,” [Online]. Available: <http://www.oracle.com/us/technologies/java/overview/index.html>. [Acesso em 18 Junho 2013].
- [35] Open Source Geospatial Foundation (OSGeo), “OpenLayers: Free Maps for the Web,” [Online]. Available: <http://openlayers.org/>. [Acesso em 18 Julho 2013].
- [36] ORACLE, “JDBC - Oracle,” [Online]. Available: <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>. [Acesso em 18 Julho 2013].
- [37] Microsoft, “XML for Analysis (XMLA),” [Online]. Available: [http://msdn.microsoft.com/en-us/library/ms187178\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms187178(v=sql.90).aspx). [Acesso em 25 07 2013].
- [38] ORACLE, “Java EE at a Glance,” [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>. [Acesso em 18 Julho 2013].
- [39] ORACLE, “Java Persistence API,” [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>. [Acesso em 25 07 2013].
- [40] “olap4j: Open Java API for OLAP,” [Online]. Available: <http://www.olap4j.org/>. [Acesso em 18 Julho 2013].
- [41] Microsoft, “MDX,” [Online]. Available: [http://msdn.microsoft.com/en-us/library/aa216767\(v=sql.80\).aspx](http://msdn.microsoft.com/en-us/library/aa216767(v=sql.80).aspx). [Acesso em 25 07 2013].
- [42] Vivid Solutions Inc., “JTS Topology Suite - Vivid Solutions Inc.,” [Online]. Available: <http://www.vividsolutions.com/jts/JTSHome.htm>. [Acesso em 18 Julho 2013].

- [43] Tribunal de Contas da União - TCU, "Portal do Tribunal de Contas da União," [Online]. Available: <http://portal2.tcu.gov.br/TCU>. [Acesso em 18 Julho 2013].
- [44] M. Y. B. S. R. Salehi, "A Formal Conceptual Model and Definition Framework for Spatial Datacubes," *Geomatica*, vol. 64, n. 3, pp. 313-326, 2010.
- [45] C. V. Ruiz e V. C. Times, "A Taxonomy of SOLAP Operators," em *XXIV Simpósio Brasileiro de Banco de Dados, 05-09 de Outubro, Fortaleza, Ceará, Brasil, Anais*, 2009.
- [46] G. Viswanathan e M. Schneider, "On the Requirements for User-Centric Spatial Data Warehousing and SOLAP," em *Database Systems for Adanced Applications*, vol. 6637, J. Xu, G. Yu, S. Zhou e R. Unland, Eds., Springer Berlin Heidelberg, 2011, pp. 144-155.
- [47] P. Marchand, A. Brisebois, Y. Bédard e G. Edwards, "Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, n. 1-2, pp. 6-20, 2004.
- [48] L. I. Gómez, S. A. Gómez e A. A. Vaisman, "A generic data model and query language for spatiotemporal OLAP cube analysis," em *Proceedings of the 15th International Conference on Extending Database Technology*, New York, NY, USA, 2012.
- [49] The Apache Software Foundation, "Apache Maven Project," [Online]. Available: <http://maven.apache.org/>. [Acesso em 18 Julho 2013].
- [50] "OpenLayers: Home," [Online]. Available: <http://openlayers.org/>. [Acesso em 15 Junho 2013].

Apêndice A

Neste apêndice é apresentado o pseudo-código responsável por traduzir uma consulta visual para a linguagem MDX. Para que ocorra a tradução e execução da consulta visual, é necessário que haja pelo menos uma hierarquia no eixo “Colunas”. Um objeto representando uma consulta MDX é gerado para cada hierarquia presente no eixo “Camadas” da consulta visual.

Ao realizar o *parsing* da consulta MDX, cada eixo da consulta será, por padrão, não nulo e ordenado de acordo com as hierarquias presentes no eixo, para isso, as seguintes funções MDX são utilizadas: NON EMPTY e HIERARCHIZE. Caso o eixo possua mais de uma hierarquia, a função MDX CROSSJOIN é aplicada. Por fim, os membros serão selecionados por meio de uma restrição MDX EXCEPT ou por um SET de membros. Após executar as consultas, é realizado um merge dos resultados.

Código 5 – Pseudo-código para a tradução da consulta visual para a linguagem de consulta MDX

```

traduzirConsultaVisual(consultaVisual){
  Tradução tradução = nova Tradução();
  Se consultaVisual possuir hierarquias no eixo Colunas{
    Para cada hierarquia do eixo Camadas da consultaVisual{
      ConsultaMDX mdx =
        traduzirParaMDX(consultaVisual, hierarquia);
      tradução.adicionaMDX(mdx);
    }
  }
  Retorna tradução;
}

traduzirParaMDX(consultaVisual, hierarquia){
  ConsultaMDX mdx = nova ConsultaMDX();

  adicionaEixoAConsultaMDX(mdx, "COLUNAS", consultaVisual);
  adicionaEixoAConsultaMDX(mdx, "LINHAS", consultaVisual);
  adicionaEixoAConsultaMDX(mdx, "PAGINAS", consultaVisual);

  adicionaCamadaAConsultaMDX(mdx, hierarquia, consultaVisual);

  adicionaFiltrosAConsultaMDX(mdx, consultaVisual);

  Retorna mdx;
}

adicionaEixoAConsultaMDX(mdx, eixo, consultaVisual){
  MDXEixo eixoLivre = mdx.recuperarPróximoEixoLivre();

  Para cada hierarquia do eixo informado da consultaVisual{

```



```

MDXHierarquia mdxHierarquia =
    nova MDXHierarquia(hierarquia.nomeÚnico);
Nivel nível =
    consultaVisual.recuperarÚltimoNívelUtilizado(hierarquia);
MDXNível mdxNível = novo MDXNível(nível.nomeÚnico);
MDXExcept except = nova MDXExcept(
    consultaVisual.recuperarEstadoDoNível(nível).
    recuperarMembrosDesselecionados());
mdxNível.adicionarRestrição(except);
mdxHierarquia.adicionarNível(mdxNível);
mdx.adicionarHierarquiaAoEixo(eixoLivre, mdxHierarquia);
}
}

adicionaCamadaAConsultaMDX(mdx, hierarquia, consultaVisual){
    MDXEixo eixoLivre = mdx.recuperarPróximoEixoLivre();
    MDXHierarquia mdxHierarquia =
        nova MDXHierarquia(hierarquia.nomeÚnico);
    Nivel nível =
        consultaVisual.recuperarÚltimoNívelUtilizado(hierarquia);
    MDXExcept except = nova MDXExcept(
        consultaVisual.recuperarEstadoDoNível(nível).
        recuperarMembrosDesselecionados());
    mdxNível.adicionarRestricao(except);
    mdxHierarquia.adicionarNível(mdxNível);
    mdx.adicionarHierarquiaAoEixo(eixoLivre, mdxHierarquia);
}

adicionaFiltrosAConsultaMDX(mdx, consultaVisual){
    Para cada hierarquia do eixo Filtros da consultaVisual{
        MDXHierarquia mdxHierarquia =
            nova MDXHierarquia(hierarquia.nomeÚnico);
        Nivel nível =
            consultaVisual.recuperarÚltimoNívelUtilizado(hierarquia);
        MDXNível mdxNível = novo MDXNível(nível.nomeÚnico);
        MDXSet set = nova MDXSet(
            consultaVisual.recuperarEstadoDoNível(nível).
            recuperarMembrosSelecionados());
        mdxNível.adicionarRestricao(set);
        mdxHierarquia.adicionarNível(mdxNível);
        mdx.adicionarHierarquiaACláusulaWhere(mdxHierarquia);
    }
}

```

As consultas na linguagem MDX, ilustradas no Código 6, foram geradas pelo pseudo-código apresentado neste apêndice. Na Figura 73 é ilustrado um exemplo de consulta visual em que foram adicionados os níveis “DESCRIÇÃO TIPO EMPENHO” e “DESCRIÇÃO SUBFUNÇÃO” no eixo “Linhas”; a medida “VALOR EMPENHO” no eixo “Colunas”; o nível “ANO” no eixo “Filtros”; e os níveis “NOME MUNICÍPIO” e “NOME MICRORREGIÃO” no eixo “Camadas”. Para o nível ano, apenas o membro “2012” foi selecionado.

Figura 73 - Consulta visual com os níveis “DESCRIÇÃO TIPO EMPENHO” e “DESCRIÇÃO SUBFUNÇÃO” no eixo “Linhas”, o nível de medidas no eixo “Colunas”, o nível “ANO” no eixo “Filtros” e os níveis “NOME MUNICÍPIO” e “NOME MICRORREGIÃO” no eixo “Camadas”



Código 6 – Consultas MDX geradas pelo pseudo-código ilustrado no Código 5 para a consulta visual ilustrada na Figura 73

```
SELECT
NON EMPTY Hierarchize(Except([Measures].[MeasuresLevel].AllMembers,
{[Measures].[MEDIA VALOR EMPENHO], [Measures].[QUANTIDADE EMPENHOS]})) ON
COLUMNS,
NON EMPTY Hierarchize(CrossJoin(Except([DIM TIPO EMPENHO].[DESCRIÇÃO TIPO
EMPENHO].[DESCRIÇÃO TIPO EMPENHO].AllMembers, {}),
Except([DIM SUBFUNCAO].[DESCRIÇÃO SUBFUNCAO].[DESCRIÇÃO
SUBFUNCAO].AllMembers, {}))) ON ROWS,
NON EMPTY Hierarchize(Except([DIM UNIDADE ORCAMENTARIA].[NOME
MUNICÍPIO].[NOME MUNICÍPIO].AllMembers,{})) ON PAGES
FROM [FAT_EMPENHO]
WHERE {[DIM TEMPO].[ANO].&[2012]}
```

```
SELECT
NON EMPTY Hierarchize(Except([Measures].[MeasuresLevel].AllMembers,
{[Measures].[MEDIA VALOR EMPENHO], [Measures].[QUANTIDADE EMPENHOS]})) ON
COLUMNS,
NON EMPTY Hierarchize(CrossJoin(Except([DIM TIPO EMPENHO].[DESCRIÇÃO TIPO
EMPENHO].[DESCRIÇÃO TIPO EMPENHO].AllMembers, {}),
Except([DIM SUBFUNCAO].[DESCRIÇÃO SUBFUNCAO].[DESCRIÇÃO
```

```
SUBFUNCAO].AllMembers, {}))) ON ROWS,  
NON EMPTY Hierarchize(Except([DIM UNIDADE ORCAMENTARIA].[NOME  
MICROREGIÃO].[NOME MICROREGIÃO].AllMembers, {})) ON PAGES  
FROM [FAT_EMPENHO]  
WHERE {[DIM TEMPO].[ANO].&[2012]}
```

Anexo A

UNIDADE ORÇAMENTÁRIA

“Constitui unidade orçamentária o agrupamento de serviços subordinados ao mesmo órgão ou repartição a que serão consignadas dotações próprias”. (Artigo 14 da Lei nº 4.320/64)

“A repartição da administração direta que o orçamento da união ou, mais especificamente, o QDD, consigna dotações específicas para a realização de seus programas de trabalho e sobre os quais essa repartição exerce o poder de disposição”. (IN/DTN nº 10, de 02 de outubro de 1991)

UNIDADE ADMINISTRATIVA

“A repartição pública da administração direta não contemplada nominalmente no orçamento da união e que depende de descentralização externa ou descentralização interna para executar seus programas de trabalho”. (IN/DTN nº 10/91)

UNIDADE GESTORA

“A unidade orçamentária ou administrativa investida de poder para gerir créditos orçamentários e/ou recursos financeiros”. (IN/DTN nº 10/91)

UNIDADE GESTORA EXECUTORA

“Unidade gestora que utiliza o crédito recebido da unidade gestora responsável. A unidade gestora que utiliza seus próprios créditos passa a ser ao mesmo tempo unidade gestora executora e unidade gestora responsável”. (IN/DTN nº 10/91)

UNIDADE GESTORA RESPONSÁVEL

Unidade gestora que responde pela realização de parte do programa de trabalho contida num crédito.

ORDENADOR DE DESPESA

“Ordenador de despesa é toda e qualquer autoridade de cujos atos resultarem emissão de empenho, autorização de pagamento, suprimento ou dispêndio”. (§ 1º do art. 80 do DL 200/67)

“Autoridade com atribuições definidas em ato próprio, entre as quais as de movimentar créditos orçamentários, empenhar despesa e efetuar pagamentos”. (IN/DTN nº 10/91)

LEI ORÇAMENTÁRIA

Lei especial que contém a discriminação da receita e despesa pública, de forma a evidenciar a política econômico-financeira e o programa de trabalho do governo, obedecidos os princípios da unidade, universalidade e anualidade.

Também chamada de LOA e de “Lei de Meios”, porque ela possibilita os meios necessários para o desempenho da função pública.

QUADRO DE DETALHAMENTO DA DESPESA - QDD

É o instrumento que detalha, a nível operacional, os projetos e atividades constantes do orçamento de um determinado exercício, especificando os elementos de despesa e respectivos desdobramentos.

“Documento que indica, por ministério/órgão e em cada unidade orçamentária, a cotização dos elementos de despesa pelos projetos e/ou atividades, podendo ter sua dotação dividida por mais de um elemento de despesa”. (IN/DTN nº 10/91)

EXECUÇÃO ORÇAMENTÁRIA

“A execução orçamentária poderá processar-se mediante a descentralização de créditos entre unidades gestoras de um mesmo órgão/ministério ou entidades integrantes dos orçamentos fiscal e da seguridade social, designando-se esse procedimento de descentralização interna.

A descentralização entre unidades gestoras de órgão/ministério ou entidades de estruturas diferentes, designar-se-á descentralização externa”. (art. 2º e parágrafo único do Decreto nº 825, de 28 de maio de 1993, que estabelece normas para a programação e execução orçamentária e financeira)

EMPENHO DA DESPESA

“O empenho da despesa é o ato emanado de autoridade competente que cria para o estado obrigação de pagamento pendente ou não de implemento de condição”. (art. 58 da lei nº 4.320/64)

“É vedado a realização de despesa sem prévio empenho”. (art. 60 da Lei nº 4.320/64).

“Será feito por estimativa o empenho da despesa cujo montante não se possa determinar”. (§ 2º do art. 60 da Lei nº 4.320/64)

“É permitido o empenho global de despesas contratuais e outras sujeitas a parcelamento”. (§ 3º do art. 60 da Lei nº 4.320/64)

“Para cada empenho será extraído um documento denominado ‘Nota de Empenho’ que indicará o nome do credor, a especificação e a importância da despesa, bem como a dedução dessa do saldo da dotação própria”. (art. 61 da Lei 4.320/64)

“O empenho será formalizado no documento ‘Nota de Empenho’, do qual constará o nome do credor, a especificação e a importância da despesa, bem como os demais dados necessários ao controle da execução orçamentária e o acompanhamento da programação financeira”. (IN/DTN nº 10/91)

“Os créditos serão utilizados mediante empenho

:

***ordinário** – quando se tratar de despesa de valor determinado e o pagamento deva ocorrer de uma só vez;*

***estimativo** – quando se tratar de despesa cujo montante não se possa determinar;*

***global** – quando se tratar de despesa contratual e outra de valor determinado, sujeitas a parcelamento”. (IN/DTN nº 10/91)*

O empenho importa deduzir seu valor da dotação adequada à despesa a realizar, por força do compromisso assumido.

O valor do empenho não poderá exceder o saldo da dotação.

São finalidades do empenho:

- firmar um compromisso. Por isso é sempre prévio em relação à despesa;
- dar garantia de que os recursos utilizados serão apropriados às despesas, pois dele consta da classificação orçamentária;
- assegurar que o crédito próprio comporte a despesa. Depois da sua emissão, o saldo disponível para assumir novos compromissos fica diminuído de seu valor;
- servir de base à liquidação da despesa;
- contribuir para assegurar a validade dos contratos, convênios e outros ajustes financeiros, mediante sua indicação obrigatória em uma das cláusulas essenciais desses termos.

LIQUIDAÇÃO

“A realização da despesa compreende, além do empenho, duas etapas finais: liquidação – quando se comprova se o credor cumpriu as obrigações objeto do empenho visando apurar:

- a origem e o objeto do que se deve pagar;
- a importância exata a pagar; e
- a quem se deve pagar”. (IN/DTN nº 10/91)

“A liquidação da despesa – inclusive daquela inscrita em restos a pagar – deverá ser feita quando do recebimento do material, da execução da obra ou da prestação do serviço, com base:

- no contrato, convênio, acordo ou ajuste, se houver;
- no empenho da despesa;
- em nota fiscal ou documento equivalente;

- no termo circunstanciado do recebimento definitivo, no caso de obra ou serviço de valor superior a ~~€z\$ 350.000,00~~ (trezentos e cinquenta mil cruzados) e equipamento de grande vulto, ou em recibo, nos demais casos”. (IN/DTN nº 10/91)

“A liquidação da despesa consiste na verificação do direito adquirido pelo credor, tendo por base os títulos e documentos comprobatórios do respectivo crédito.

Essa verificação tem por fim apurar:

I – a origem e o objeto do que se deve pagar;

II – a importância exata a pagar;

III – a quem se deve pagar a importância, para extinguir a obrigação”. (art. 63 e § 1º da Lei nº 4.320/64)

PAGAMENTO

Terceiro estágio da despesa pública.

Caracteriza-se pela emissão da ordem bancária em favor do credor.

“A ordem de pagamento é o despacho exarado por autoridade competente, determinando que a despesa seja paga”. (art. 64 da Lei nº 4.320/64)

“Pagamento – Quando comprovado o direito adquirido pelo credor e verificada a existência de recursos suficientes, extinguindo-se a obrigação”. (IN/DTN nº 10/91)

“O pagamento da despesa só poderá ser efetuado, após sua regular liquidação, mediante ordem bancária de crédito ou de pagamento, facultado ao credor escolher a forma que melhor lhe convier”. (IN/DTN nº 10/91)

“A autorização para pagamento compete ao ordenador de despesa, que poderá delegar essa autorização”. (IN/DTN nº 10/91)

RESTOS A PAGAR

“Consideram-se restos a pagar as despesas empenhadas mas não pagas até o dia 31 de dezembro, distinguindo-se as processadas das não-processadas”. (art. 36 da Lei nº 4.320/64)

Serão automaticamente inscrita em restos a pagar, no encerramento do exercício, as despesas empenhadas:

- liquidadas e não pagas;
- não liquidadas, desde que:

1 - vigente o prazo para cumprimento da obrigação assumida pelo credor, indicado na nota de empenho;

2 - vencido o prazo de que trata o subitem anterior, mas esteja em curso a liquidação da despesa ou seja de interesse da administração exigir cumprimento da obrigação assumida pelo credor;

3 - se destinem a atender transferências a instituições públicas ou privadas;

4 - correspondam a compromissos assumidos no exterior.

A inscrição de despesa em restos a pagar terá validade até 31 de dezembro do ano subseqüente, quando será automaticamente cancelada. Permanecerá em vigor, no entanto, o direito do credor por cinco anos, a partir da data de inscrição.

DESPESA DE EXERCÍCIOS ANTERIORES

“As despesas de exercícios anteriores encerrados, para as quais o orçamento respectivo consignava crédito próprio, com saldo suficiente para atendê-las, que não se tenham processado na época própria, bem como os restos a pagar com prescrição interrompida e os compromissos reconhecidos após o encerramento do exercício correspondente, poderão ser pagos à conta de dotação específica consignada no orçamento, discriminada por elementos, obedecida, sempre que possível, a ordem cronológica”. (Art. 37 da Lei nº 4.320/64)

“Os compromissos decorrentes de obrigação de pagamento criada em virtude de lei e reconhecidos após o encerramento do exercício”. (IN/DTN nº 10/91)

O reconhecimento da dívida a ser paga a conta de despesas de exercícios anteriores cabe à autoridade competente para empenhá-la, devendo o processo conter, no mínimo, os seguintes elementos:

- importância a pagar;
- nome, CPF ou CGC e endereço do credor;
- data do vencimento do compromisso;
- causa da inobservância do empenho, se for o caso.

“A autorização de pagamento de despesas de exercícios anteriores deverá ser dada no próprio processo de reconhecimento de dívida”. (IN/DTN nº 10/91)

SUPRIMENTO DE FUNDOS

Em casos excepcionais, a autoridade ordenadora poderá autorizar o pagamento da despesa por meio de suprimento de fundos, quando essa não puder ser realizada pelo processo normal da execução orçamentária, assim considerada nos seguintes casos:

- para atender em viagens ou serviços especiais que exijam pronto pagamento em espécie;
- quando a despesa deva ser feita em caráter sigiloso, conforme se classificar em regulamento e constar do ato de concessão;
- para atender despesa de pequeno vulto, assim entendida aquela cujo valor não ultrapassar a R\$ 200,00, no caso de compras e serviços, e a ~~cinquenta-MVR~~, no caso de obras.

O Suprimento de Fundos consiste na entrega do numerário a servidor, sempre precedida de empenho na dotação própria da despesa a realizar será feita mediante:

- ordem bancária de pagamento em favor do suprido; ou
- crédito em conta bancária em nome do suprido, aberta, com autorização do ordenador de despesa, para esse fim.

A fixação do valor do suprimento de fundos ficará a critério do ordenador de despesa.

Não poderá ser concedido suprimento de fundos:

- a responsável por dois suprimentos;
- a servidor que tenha a seu cargo a guarda ou a utilização do material a adquirir, salvo quando não houver na repartição outro servidor;
- a responsável por suprimento de fundos que não tenha prestado contas de sua aplicação no prazo previsto;
- a servidor declarado em alcance ou que esteja respondendo a inquérito administrativo.

A prestação de contas da aplicação dos recursos oriundos de suprimento de fundos deverá ser feita mediante apresentação dos seguintes documentos:

- cópia do ato de concessão do suprimento;
- primeira via da nota de empenho da despesa;
- extrato da conta bancária, se for o caso;
- demonstração de receitas e despesas;
- comprovante de recolhimento do saldo, se for o caso.
- comprovantes das despesas realizadas, devidamente atestados e emitidos em data igual ou posterior à da entrega do numerário e em nome do órgão onde o suprido esteja em exercício, a saber:

1 - no caso de compra de material: nota fiscal de venda ao consumidor;
2 - no caso de prestação de serviços por pessoa jurídica: nota fiscal de prestação de serviços;

3 - no caso de prestação de serviços por pessoa física: recibo comum - se o credor não for inscrito no INSS, e recibo de pagamento de autônomo (RPA) - se o credor for inscrito no INSS.

(ver arts. 68 e 69 da Lei nº 4.320/64)

CRONOGRAMA DE DESEMBOLSO

Instrumento pelo qual a unidade orçamentária programa o pagamento das despesas autorizadas na lei orçamentária.

Previsão dos pagamentos tendo em vista assegurar a entrega, aos diversos órgãos da administração pública federal, de forma automática e em tempo hábil, dentro dos limites previamente estabelecidos, dos recursos financeiros necessários para a execução de seus programas de trabalho.

Elaboração da programação de utilização de recursos realizada pelos órgãos setoriais, que evidencia as necessidades para o desenvolvimento dos respectivos programas, em cotejo com o esperado comportamento da receita do Tesouro.

MOVIMENTAÇÃO DOS RECURSOS FINANCEIROS

Denomina-se liberação de cota a parcela que o órgão central de programação financeira autorizará o Banco do Brasil S/A, na qualidade de agente financeiro do Tesouro Nacional, a colocar à disposição dos usuários.

Chama-se Repasse a importância que uma unidade orçamentária transfere a outra unidade orçamentária de outro ministério ou órgão.

O repasse também será utilizado pelo órgão setorial de programação financeira para transferir recursos às entidades da administração indireta.

A figura do repasse financeiro está associada à descentralização externa.

Intitula-se sub-repasse a importância que uma unidade orçamentária transfere a outra unidade orçamentária ou administrativa do mesmo ministério ou órgão. A figura do sub-repasse está ligada à descentralização interna.

FLUXOGRAMA DA EXECUÇÃO ORÇAMENTÁRIA E FINANCEIRA

