

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA – INE
CIÊNCIAS DA COMPUTAÇÃO

Uma aplicação OLAP com visualização cartográfica via Web

Florianópolis, 2007.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA – INE
CIÊNCIAS DA COMPUTAÇÃO

Uma aplicação OLAP com visualização cartográfica via Web

Miguel Soares

Nuno Santos

Trabalho de Conclusão do Curso apresentado como um dos requisitos para
obtenção do grau de Bacharel em Ciências da Computação.

Florianópolis, SC

2008/1

Miguel Soares

Nuno Santos

Uma aplicação OLAP com visualização cartográfica

Trabalho de conclusão de curso apresentado como parte dos requisitos para
obtenção do grau de Bacharel em Ciências da Computação.

Orientador: Renato Fileto

Banca examinadora:

Angelo Augusto Frozza

Ronaldo dos Santos Mello

Índice

1	Introdução.....	10
2	Fundamentos	13
2.1	Data Warehouse	13
2.2	O modelo dimensional.....	14
2.2.1.1	Esquema Estrela (Star Schema).....	16
2.2.1.2	Esquema floco de neve (Snow Flake Schema).....	17
2.2.1.3	Comparação entre os dois esquemas	19
2.3	OLAP.....	20
2.3.1	Padrões para conexão com ferramentas OLAP.....	21
	XML – eXtensible Markup Language	21
	XMLA – XML for Analysis.....	23
	MDX – Multidimensional Expressions	25
2.4	Sistemas de Informação Geográfica	26
2.4.1	Técnicas de GIS	26
	Coleta de Dados.....	26
	Matriz.....	26
	Vetor.....	27
	Vantagens e Desvantagens.....	28
2.4.2	Componentes da arquitetura de um GIS	28
3	Análise de ferramentas	29
3.1	Microsoft Visual Studio.....	29
3.2	Servidor OLAP	30
3.2.1	Microsoft SQL Server 2005 Analysis Services.....	30
3.2.2	Pentaho Analysis Services: Mondrian.....	36
3.3	Interface OLAP.....	37
3.3.1	Desenvolvimento de raiz.....	37
	Implementação.....	37
3.3.2	Microsoft SQL Reporting Services	40
	Os diferentes tipos de relatório:.....	40
	Características do Microsoft SQL Server Reporting Services [MICROSOFT, 2008]:.....	41
	A ferramenta Implementada:	42
3.4	Ferramentas GIS.....	44
3.4.1	Microsoft MapPoint 2006 North America	44
3.4.2	MapWindow GIS	46
3.4.3	ArcGIS Explorer	47
3.4.4	AvisMap GIS Engine.....	47
3.4.5	Quantum GIS.....	48
3.4.6	Comparativo	49
4	Arquitetura proposta.....	51
5	Implementação	54
5.1	MapWindow	54
5.1.1	Gerar mapas.....	54
5.1.2	Servidor de mapas	58
5.2	WebService	58

6	Trabalhos relacionados	61
7	Conclusões e trabalhos futuros	63
	Referências Bibliográficas	65

Lista de figuras

Figura 1 - Representação gráfica de um Data Warehouse [TRIGUEIROS, 2006]	13
Figura 2 - Modelo Estrela [SHIGUNOV, 2007].....	17
Figura 3 - Floco de neve.....	18
Figura 4 - Exemplo do modelo Snow Flake.....	19
Figura 5 - Representação de um cubo OLAP.....	21
Figura 6 - Representação matricial.....	27
Figura 7 - Representação vectorial	27
Figura 8 - Interação das ferramentas nas diferentes camadas de BI [NORTHRIDGE SYSTEMS, 2008].....	31
Figura 9 – Lado do DW e localização do Analysis Services.....	32
Figura 10 - Conexão via ODBC	34
Figura 11 - Criação do StarScheme no Analysis Services.....	35
Figura 12 - Wizard de criação do Cubo.....	36
Figura 13 - Resultado de uma consulta	39
Figura 14 - Exemplo de relatório usando o Reporting Services.....	44
Figura 15 - Quadro comparativo das ferramentas GIS estudadas	50
Figura 16 - Arquitectura do site VAOC	52
Figura 17 - Resultado de uma <i>query</i> por país.....	57
Figura 18 - Protótipo de aplicação de mapas	58
Figura 19 - Relação do Webservice com os outros componentes.....	60

Lista de Quadros

Quadro 1 - Diferenças entre modelos na construção de DW [MELLO, 2002].....	15
Quadro 2 - Código XML descrevendo um curriculum vitae.....	23
Quadro 3 - Exemplo de um XMLA usando o método Discover	24
Quadro 4 - Consulta MDX	25
Quadro 5 - Objeto de comunicação ADOMD	33
Quadro 6 - Arquivo WebConfig.xml	33
Quadro 7 - Comparação de servidores OLAP.....	37
Quadro 8 - Organização dos dados da tabela	38
Quadro 9 - Exemplo de uma consulta MDX.....	43
Quadro 10 - Características dos componentes	52
Quadro 11 - Processo de pesquisa do arquivo geográfico.....	55
Quadro 12 - Método de colorir de cada polígono	56

Lista de siglas e abreviações

API – Application Programming Interface

BI – Business Intelligence

CAD – *Computer Aided Design*

DW – Data Warehouse

GIS – Geographic Information Systems

GMLA – *GML for Analysis*

GOLAPA – *Geographic Online Analytical Processing Architecture*

IDE – *Integrated Development Environment*

MDX – *Multidimensional expressions*

OLAP – Online Analytical Processing

SOAP – Simple Object Access Protocol

UFSC – Universidade Federal de Santa Catarina

XML – eXtensible Markup Language

XMLA – *eXtensible Markup Language for Analysis*

Resumo

Nos dias de hoje a informação assume grande importância na sociedade, e especialmente nas empresas. Cada vez mais, as empresas procuram guardar todos os dados para que a partir daí consigam gerar informação, no entanto, volumes imensos de dados muitas vezes dificultam a recolha de informação eficaz, interessante e de fácil percepção. Torna-se assim um desafio encontrar ferramentas que transformem grandes quantidades de dados em poucas informações, mas úteis e importantes.

Este trabalho integra funcionalidades de processamento analítico *online* (OLAP) e Sistemas de Informação Geográfica (GIS), com o objetivo de gerar mapas para visualizar geograficamente as tabelas de dados produzidas através do OLAP. O protótipo desenvolvido, que inclui uma interface de usuário na *Web*, utiliza a ferramenta MapWindow GIS para gerar os mapas e o Microsoft Analysis Services acoplado ao SQL Server 2005 como servidor OLAP. A comunicação entre a interface OLAP e o *MapWindow* é feita através de um Web Service que encapsula as funcionalidades do MapWindow necessárias para gerar os mapas.

Palavras-chave: GIS, OLAP, Analysis Services, MapWindow, Web.

Abstract

Nowadays information assumes big importance in society and specially companies. More and more, companies save every data they can, so that they can generate information trough there, however huge amount of data, lots of times, difficult the gathering of an effective, interesting and of easy perception information. This way is a challenge to find tools that transform big quantity of data into few information, but useful and important.

This work integrates functionalities of On-Line Analytical Processing (OLAP) and Geographic Information Systems (GIS), in order to generate maps for geographic visualization of data tables produced with OLAP. The developed prototype, which includes a user interface on the Web, uses the MapWindow GIS tool to generate the maps and the Microsoft Analysis Services coupled to SQL Server 2005 as the OLAP server. The communication between the OLAP interface and MapWindow is made trough a Web Service, that encapsulates the MapWindow functionality necessary to generate the maps.

Keywords: GIS, OLAP, Analysis Services, MapWindow, Web.

1 Introdução

O uso de ferramentas OLAP (*On-Line Analytical Processing*) [ANOZELO, 2004; OLAP, 2007; TRIGUEIROS 2006] tem se difundido nos últimos anos, graças à disponibilidade de soluções proprietárias de baixo custo e de ferramentas de código aberto¹. Entre as soluções proprietárias destaca-se o Microsoft Analysis Services, distribuído com o SQL-Server. Entre as ferramentas de código aberto para a implementação de sistemas baseados em OLAP com interface Web encontram-se o SpagoBI, o OpenI e o Pentaho, todas baseadas na biblioteca JPivot [MACDONALD 2007]. Essas ferramentas permitem a implementação de sistemas baseados em OLAP com sofisticadas interfaces sobre a *Web*, que possibilitam inclusive a geração de gráficos.

Muitas aplicações de OLAP envolvem dados geográficos, mas as ferramentas OLAP atuais não se integram facilmente com sistemas de informação geográfica (GIS – *Geographical Information Systems*). Devido a esta lacuna, a integração de GIS com OLAP é tema de estudo por parte de variados grupos de pesquisa. Um deles é o grupo da Universidade Federal de Pernambuco, que desenvolveu a arquitetura GOLAPA (*Geographic Online Analytical Processing Architecture*) [FIDALGO 2004] para a integração entre OLAP e GIS. GOLAPA inclui cinco camadas e utiliza a linguagem GMLA (*GML for Analysis*), proposta pelo mesmo grupo, para o intercâmbio de informação entre os módulos, de modo a permitir a geração de mapas referentes a visões tabulares de dados produzidas com OLAP.

Outra forma de integração de OLAP com GIS existe com o SOLAP (Spatial OLAP) [RIVEST 2005] onde é definida uma nova linguagem para a manipulação dos dados diretamente no Data Warehouse. DAMIANI [2004] por sua vez defende um modelo

¹ A diferença fundamental entre Código Aberto e Software Livre acontece sobretudo na parte dos valores das organizações responsáveis por cada um. Enquanto os apoiantes do Código livre consideram-no como uma metodologia de desenvolvimento, os do software livre levam-no mais longe e consideram que o software não livre é um problema social que só pode ser respondido com software livre.

dimensional espacial de dados para objetos de caráter geográfico.

O trabalho que está sendo realizado na Universidade Federal de Santa Catarina [MACDONALD 2007; SHIGUNOV 2007; GROSS 2007], por outro lado, utiliza XMLA (*eXtensible Markup Language for Analysis*) e MDX (*Multidimensional expressions*) [IME 2007; XMLA 2007] para a comunicação com servidores OLAP, de acordo com o padrão estabelecido de fato na maioria das ferramentas disponíveis atualmente para OLAP. A comunicação com servidores de mapa ainda requer estudos adicionais para definir APIs na forma de serviços Web, baseadas em padrões sendo propostos atualmente para o intercâmbio de dados geográficos, como aqueles do OpenGIS².

Este trabalho aborda a temática da integração entre OLAP e GIS do ponto de vista prático. Ele foca na utilização de ferramentas proprietárias (Microsoft), mas recorre a ferramentas *open source* quando estas são consideradas mais convenientes para o sistema. O objetivo é apresentar a informação resultante de consultas OLAP na forma de mapas, de modo a prover uma interface mais visual e facilitar a percepção de informação relevante pelo usuário. Isso ajuda a preencher a lacuna existente na área de visualização de dados multidimensionais que contêm elementos geográficos.

O protótipo desenvolvido resulta de uma pesquisa de ferramentas de *Business Intelligence* (BI) e de GIS. O projeto é dividido em duas partes distintas: o desenvolvimento de uma interface OLAP básica sobre a Web e a visualização cartográfica dos resultados de consultas OLAP, usando uma ferramenta GIS. A interface OLAP criada permite a utilização de qualquer servidor OLAP, pois a comunicação com tal servidor é feita através de XMLA, um padrão de fato. Esta característica aumenta a flexibilidade do sistema desenvolvido, pois permite utilizar tecnologias diferenciadas no servidor OLAP, consoantes com as necessidades específicas do usuário.

² Organização internacional que agrega empresas que trabalham na área de Sistemas de Informação Geográficos e que procura estabelecer padrões comuns nesta área. Pode ser acessado em: <http://www.opengeospatial.org/standards>

A segunda parte do trabalho refere-se à visualização geográfica da informação consultada via OLAP. Quando o usuário aciona a visualização de mapas para uma dada visão de dados tabulares gerada através de OLAP, tal visão é enviada, no formato XMLA, para o serviço Web (*Web Service*) de geração de mapa. Este serviço recolhe arquivos no formato ESRI shapefile referentes às representações (extensões) de entidades geográficas (por exemplo estados e cidades) referenciados em visões tabulares de dados gerados através de OLAP. O mapa é gerado acionando a ferramenta GIS MapWindow, com as extensões geográficas e as cores definidas pelo servidor de mapas, em uma escala com cores variando na mesma proporção que o valor da medida apresentada no XMLA para as respectivas entidades geográficas.

Este documento segue com uma breve análise dos fundamentos necessários para o entendimento do trabalho no capítulo 2. O capítulo 3 apresenta um estudo acerca de ferramentas disponíveis, comerciais ou não. O capítulo 4 apresenta a arquitetura proposta e o 5º apresenta alguns detalhes da sua implementação. O capítulo 6 aborda os trabalhos relacionados. Finalmente, o 7º conclui o trabalho, enumerando as contribuições e os trabalhos futuros.

2 Fundamentos

2.1 Data Warehouse

Um *data warehouse* (DW), ou armazém de dados [CARDOSO 2006; DT 2007], integra dados de diversas fontes para propiciar uma visão histórica, unificada e não volátil da informação para apoio à tomada de decisão. A informação dessas fontes é transferida para o DW e organizada para permitir acesso unificado e mais fácil. A Figura 1 apresenta uma data warehouse que através de um processo periódico de extração, transformação e carga, integra dados de várias fontes e provê visões dos dados integrados para diferentes grupos de usuários e ferramentas de análise de dados através dos Data Marts³.

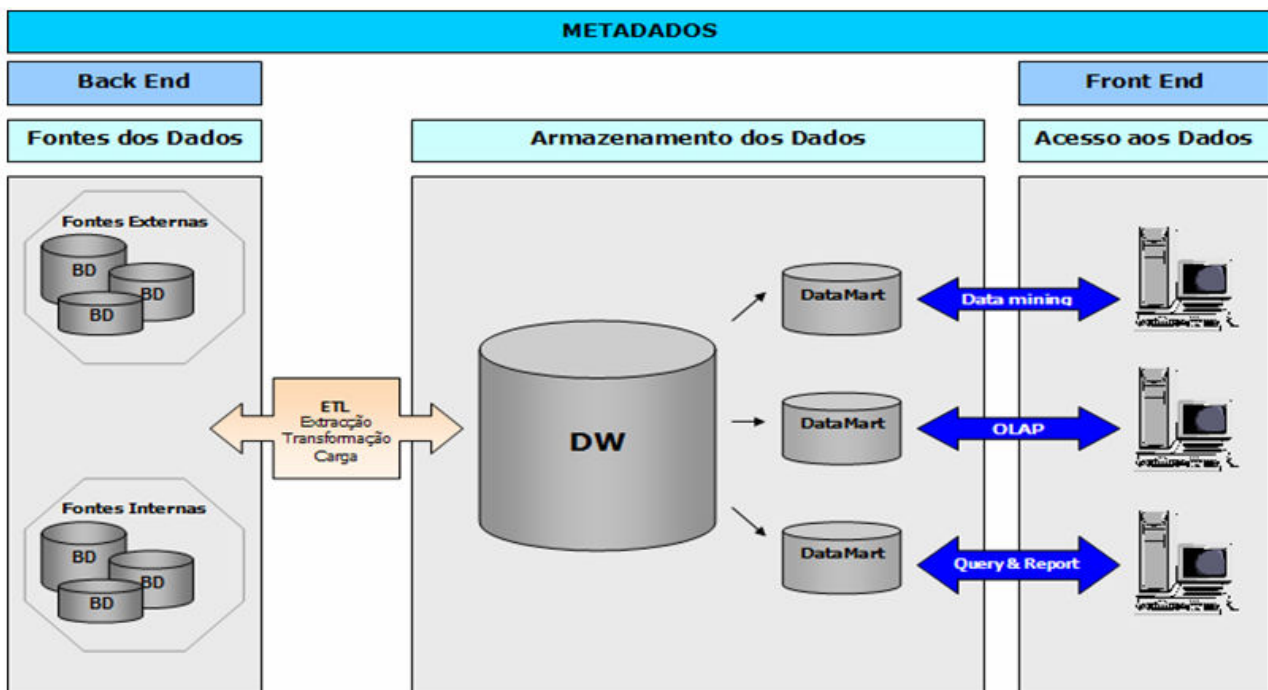


Figura 1 - Representação gráfica de um Data Warehouse [TRIGUEIROS, 2006]

O que torna o Data Warehouse uma ferramenta muito poderosa é a capacidade de reunir informações que se situam em várias fontes e reuni-las num só banco de dados

³ Data Mart é um banco de dados multidimensionais com propriedades semelhantes às de um Data Warehouse, mas mais pequeno. Podem representar uma parte do DW ou então pode ser o primeiro passo para a criação de um Data Warehouse..

sob uma representação homogênea em um só local. Este projeto utiliza o modelo dimensional para a representação unificada dos dados da DW, a mais utilizada para este tipo de base de dados.

2.2 O modelo dimensional

O modelo dimensional representa a informação num cubo multidimensional, cujas células contêm medidas (por exemplo valor vendido) e as dimensões contêm hierarquias de dados descritivos para as medidas (por exemplo espaço, tempo e produto). Este modelo é voltado para consultas analíticas. Ele permite analisar e relacionar informações de forma simples e eficaz, o que o torna muito poderoso [GOMES 2005; INMON 1997, TRIGUEIROS 2006]

O modelo dimensional é constituído por quatro componentes:

- **Fato:** é o objeto a analisar e que representa o negócio. A quantidade de fatos é que determina o modelo a usar (por exemplo crime, venda).
- **Medida:** define como mensurar um facto (por exemplo valor, quantidade)
- **Dimensões de análise dos fatos:** definem diferentes aspectos para avaliar os fatos (e.g., Quando? Onde? Quem?)
- **Hierarquias:** Organizam a informação descritiva de cada dimensão de análise dos fatos (e.g., uma hierarquia de tempo com valores para anos, meses e dias).

Data Warehouses podem ser implementados em sistemas de gerenciamento de bancos de dados com modelo dimensional nativo ou sobre bancos de dados relacionais.

As principais diferenças entre efetuar uma consulta OLAP (Modelo dimensional) e uma consulta OLTP (Modelo relacional) estão presentes no Quadro 1.

Quadro 1 - Diferenças entre modelos na construção de DW [MELLO, 2002]

	OLTP	OLAP
Função	Automatizar operações diárias	Auxiliar na tomada de decisão
Usuário Humano	Cliente, Atendente	Executivo, Analista, Eng de conhecimento
Modelo Lógico	Relacional	Multidimensional
Granularidade	Unica e Atômica	Múltipla e Agregada
Temporalidade dos dados	Apenas valor corrente atualizada constantemente	Histórico dos Valores, completado periodicamente
Consultas	Simples, pré definidas	Complexas e ad-hoc
Direção	Tanto ler como escrever	Essencialmente ler

Cada um dos modelos tem as suas vantagens e desvantagens. Quando é escolhido o modelo para gerir a base de dados é preciso ter em conta qual é o objetivo pretendido, por exemplo. Para o objectivo de retirar conclusões estruturadas e organizadas para construir um relatório, é mais vantajoso escolher o modelo dimensional, mas, se por outro lado, o objetivo é apenas retirar valor das tabelas é mais simples usar o modelo relacional.

Existem algumas diferenças entre o modelo relacional e o modelo dimensional. Neste trabalho foi concluído que o modelo dimensional é o mais adequado, pelas necessidades da manipulação dos dados para gerar relatórios. A escolha do tipo de DW depende das necessidades do usuário.

Esquemas dimensionais de data warehouses podem ser expressos no modelo relacional nos esquema estrela ou no esquema floco de neve, como descrito a seguir.

2.2.1.1 Esquema Estrela (*Star Schema*)

O estilo de esquema dimensional chamado em estrela (*star schema*), é centrado em uma tabela de facto, com múltiplas junções conectando-a a outras tabelas, sendo estas chamadas de tabelas de dimensão. Cada uma das tabelas de dimensão possui apenas uma junção com a tabela central, conforme se pode ver na Figura 2. Um esquema no estilo estrela tem a vantagem de ser simples e intuitivo, mas também faz uso da indexação e união de tabelas.

As tabelas de fatos contêm milhares ou milhões de valores e medidas do negócio da empresa, como transações de vendas ou compras. Cada uma destas medidas tem uma dimensão relacionada. Os fatos podem conter todo o tipo de dados, no entanto os que possuem maior importância na análise de informação são os fatos numéricos aditivos já que estes facilitam a geração do conjunto de respostas. Uma outra característica é o fato desta tabela não armazenar espaço vazio, ou seja, todos os campos da tabela têm que estar preenchidos.

As tabelas de dimensão armazenam as descrições das dimensões do negócio. Cada uma dessas descrições ajuda a definir um componente da respectiva dimensão. Uma das principais funções dos atributos de tabelas de dimensão é servir como fonte para restrições em uma consulta ou como cabeçalhos de linha no conjunto de resposta do usuário.

Muitos atributos de uma dimensão estão diretamente relacionados entre si e possuem uma hierarquia, que existe quando há uma relação entre diferentes dados dentro da tabela dimensão. Ao se analisar a Figura 2 é possível ver uma hierarquia na tabela candidato: estado_endereço e cidade_endereço, onde o primeiro é constituído por

muitas ocorrências do segundo.

A Figura 2 representa o modelo estrela sobre o qual o Data Warehouse do vestibular da UFSC de 2006 [SHIGUNOV 2007] foi criado.

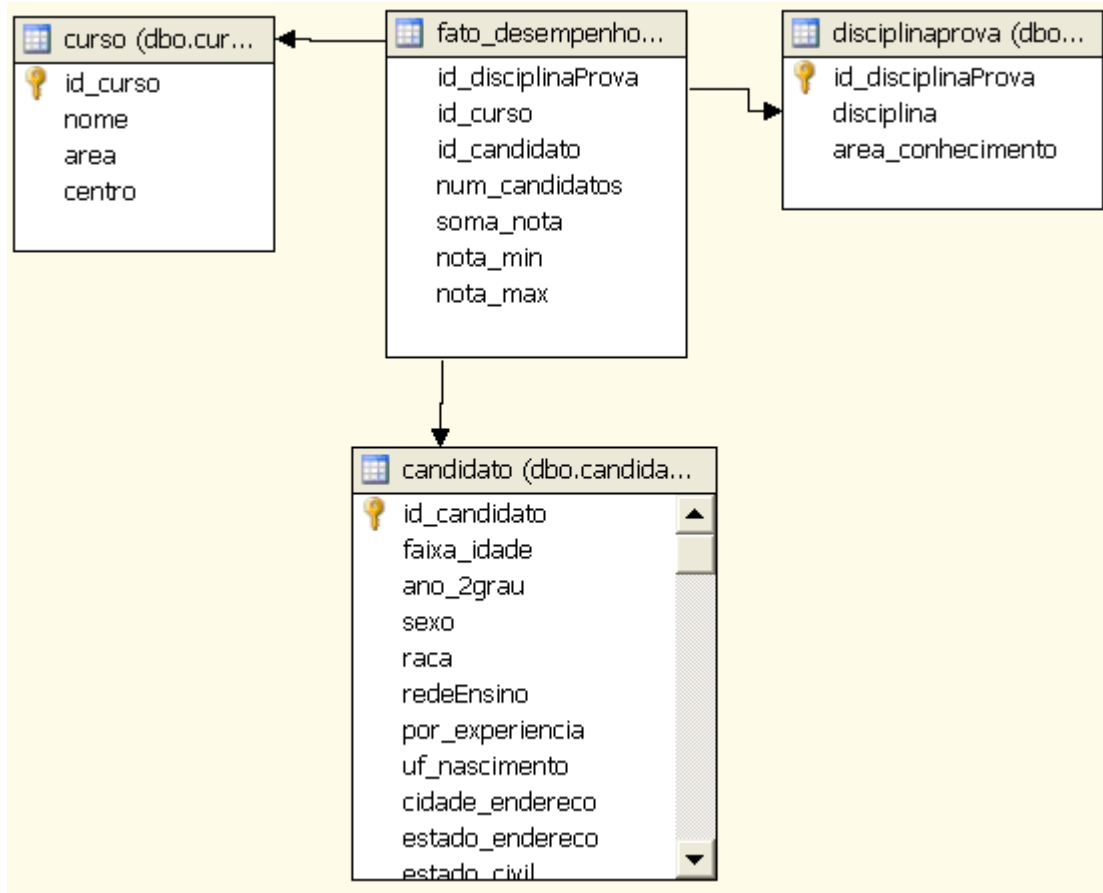


Figura 2 - Modelo Estrela [SHIGUNOV, 2007]

2.2.1.2 Esquema floco de neve (*Snow Flake Schema*)

No modelo Floco de Neve, as tabelas dimensionais relacionam-se com a tabela de fatos, mas algumas dimensões relacionam-se apenas entre elas, isto ocorre para fins de normalização das tabelas dimensionais, visando diminuir o espaço ocupado por estas tabelas.

No modelo floco de neve existem tabelas de dimensões auxiliares que normalizam as tabelas de dimensões principais. Na Figura 4 é possível verificar que as tabelas *Ano*,

Mês e *Dia* normalizam a Dimensão Tempo, e as tabelas *Categoria*, *Departamento* e *Marca*, que normalizam a Dimensão Produto.

A principal diferença entre o esquema floco de neve e o esquema estrela é o fato do primeiro possuir um maior número de tabelas para representar as mesmas dimensões e também de necessitar de menos espaço em disco com a mesma informação.

Este modelo chama-se floco de neve, pois cada dimensão se divide em várias outras tabelas, organizadas de forma que lembram um floco de neve, como se pode ver na Figura 3. No centro encontra-se a tabela fato e nas suas ramificações estão tabelas relacionadas entre si que se vão ramificando mais à medida que a granularidade dos dados vai aumentando.

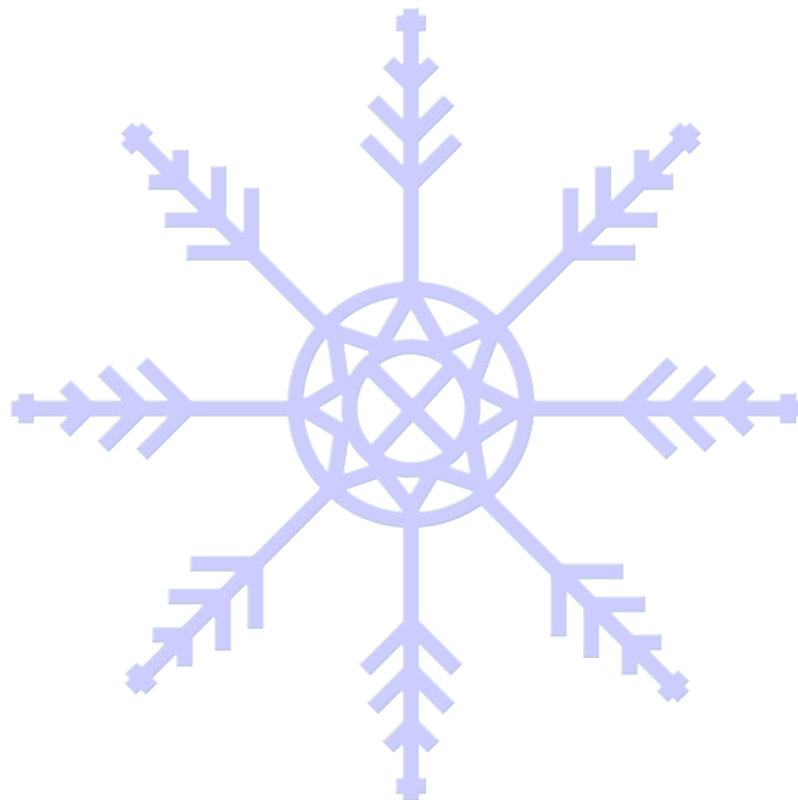


Figura 3 - Floco de neve

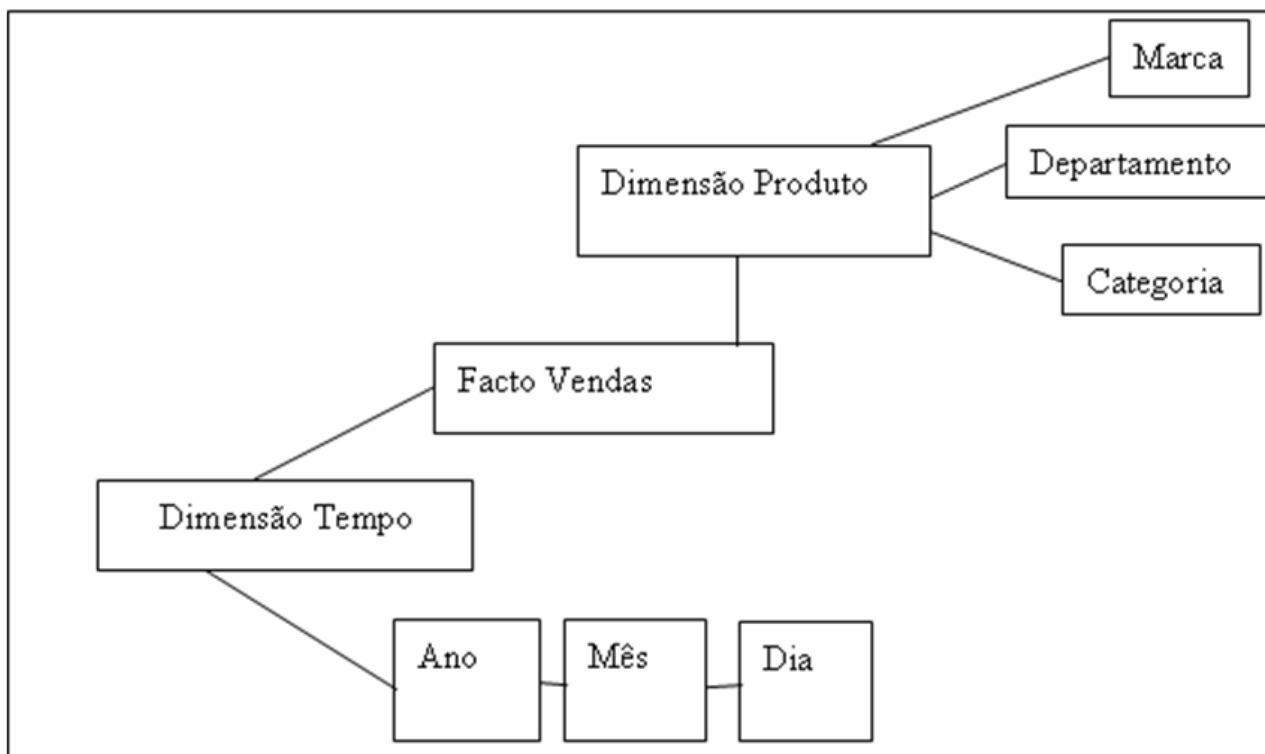


Figura 4 - Exemplo do modelo Snow Flake

2.2.1.3 Comparação entre os dois esquemas

O Modelo Floco de Neve (Snow Flake) reduz o espaço de armazenamento dos dados dimensionais mas acrescenta várias tabelas ao modelo, deixando-o mais complexo e tornando mais difícil a navegação pelos softwares que utilizam o banco de dados. Um outro fator é que é necessário utilizar mais tabelas para executar uma consulta, então mais *JOINS* de instrução SQL são feitos, tornando o acesso aos dados mais lento do que no modelo estrela.

O Modelo em Estrela (Star Schema) é mais simples e mais fácil de navegação pelos softwares, porém desperdiça espaço devido à repetição das mesmas descrições ao longo de toda a tabela. Análises feitas mostram que o ganho de espaço normalizando este esquema resulta em um ganho menor que 1% do espaço total no banco de dados, e como os custos do espaço em banco de dados são reduzidos quando comparados com

outros este problema não é considerado crítico ou muito relevante pelos arquitetos dos Data Warehouses. Assim sendo existem outros factores mais importantes para serem avaliados para redução do espaço em disco como a adição de agregados e alteração na granularidade dos dados, que o esquema em estrela consegue responder com muito sucesso.

Devido aos factores atrás descritos, o mercado adotou o modelo estrela como padrão na definição de Data Warehouses. A criação de tabelas auxiliares para dimensões é somente feita para dimensões específicas quando for estritamente necessário ou quando demonstrar um benefício que justifique a perda de desempenho nas consultas, que também não é tão grande dependendo da forma que estas tabelas são construídas e a quantidade de registros que elas contiverem.

2.3 OLAP

O processamento analítico on-line (*OLAP – Online Analytical Processing*) [ANOZELO 2004; OLAP 2007; TRIGUEIROS 2006] permite analisar informação de um cubo dimensional em vários níveis de foco e agregação. OLAP é uma tecnologia que permite ao usuário navegar nos dados do cubo e efetuar operações como drill-down (descer na hierarquia de uma dimensão do cubo e ter informação mais detalhada), roll-up (subir na hierarquia de uma dimensão, agregando dados) e drill-across (analisar outra tabela de fatos, mantendo o nível da hierarquia de dimensões compartilhadas). Por exemplo, se o cubo que armazena os dados do vestibular tiver as medidas detalhadas por município, será possível fazer um drill-down para obter os valores de uma certa medida para os municípios do estado de Santa Catarina e, posteriormente, um roll-up para obter os valores agregados para cada estado do Brasil.

A Figura 5 apresenta um cubo para armazenar dados do vestibular da UFSC, com

as dimensões Sexo e Local Geográfico. Os valores das medidas (nota mínima, nota máxima, soma das notas e número de candidatos) são armazenados em células referentes a um membro de dimensão. Por exemplo, a célula de tras no canto superior esquerdo armazena a nota mínima dos candidatos do sexo feminino provenientes do estado do Rio Grande do Sul.

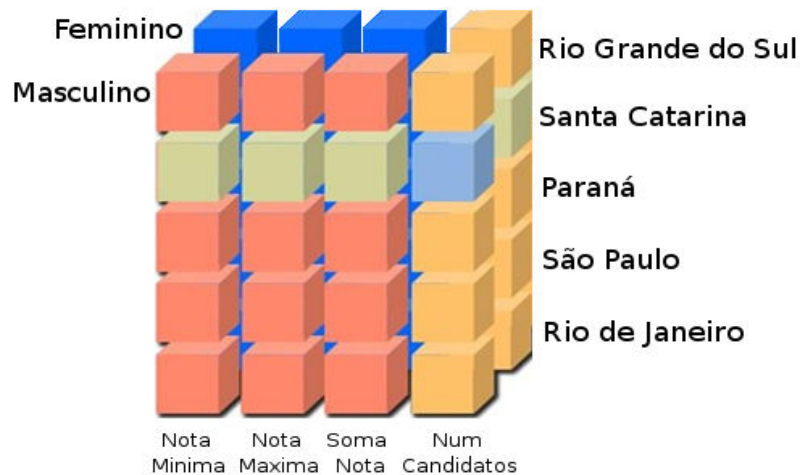


Figura 5 - Representação de um cubo OLAP

2.3.1 Padrões para conexão com ferramentas OLAP

XML – eXtensible Markup Language

A XML [HAROLD 2001] é uma especificação técnica desenvolvida pela W3C (World Wide Web Consortium – entidade que define padrões e diretrizes para a Web), para superar as limitações do HTML, que é o padrão das páginas da Web.

A linguagem XML é definida como um formato universal para dados estruturados na Web. Esses dados consistem em tabelas, desenhos, parâmetros de configuração, etc. A linguagem então trata de definir regras que permitem escrever esses documentos de forma que sejam adequadamente visíveis ao computador.

Objetivos do XML:

1. Separação do conteúdo da formatação
2. Simplicidade e legibilidade, tanto para humanos quanto para computadores
3. Possibilidade de criação de tags sem limitação
4. Criação de arquivos para validação de estrutura
5. Interligação de bancos de dados distintos
6. Concentração na estrutura da informação, e não na sua informação

Um exemplo de código XML pode ser visto no Quadro 1.

Quadro 2 - Código XML descrevendo um curriculum vitae

```
<?xml version="1.0" encoding="UTF-8"?>
<currículo>
  <InformacaoPessoal>
    <DataNascimento>02-04-88</DataNascimento>
    <Nomecompleto>Leonardo da SILVA Machado</Nomecompleto>
    <Contatos>
      <Morada>R.Caratinga, 112,Anchieta, Brasil</Morada>
      <Telefone>97446182</Telefone>
      <CorreioEletronico>leodsm01@gmail</CorreioEletronico>
    </Contatos>
    <Nacionalidade>brasileiro</Nacionalidade>
    <Sexo>M</Sexo>
  </InformacaoPessoal>
</currículo>
```

XMLA – XML for Analysis

XMLA [XMLA 2007] é uma API (Application Program Interface) baseada em XML e Web services para a troca de dados dimensionais entre um cliente e um servidor OLAP, independentemente da plataforma e da linguagem de implementação desses componentes. A principal razão para o reconhecimento de XMLA como padrão de fato é ela ser suportada pelos principais fabricantes OLAP, com exceção da ORACLE. Através

de Web Services pode-se intercambiar dados resultantes de consultas OLAP entre diferentes clientes e servidores, permitindo grande interoperabilidade e flexibilidade para a troca de dados, além de permitir a utilização de diferentes ferramentas.

O protocolo XMLA possui dois métodos que utilizam o protocolo SOAP:

Discover – Obtém informação e metadados de um Web Service. Este método permite especificar o que se está à procura, as restrições e as propriedades. Tem como resultado um conjunto de linhas.

Execute – Este método é utilizado para enviar pedidos de ações ao servidor, no qual está incluído transferência de dados (retirar e colocar dados no servidor)

Quadro 3 - Exemplo de um XMLA usando o método Discover

```
<Discover xmlns="urn:schemas-microsoft-com:xml-analysis">
  <RequestType>MDSHEMA_CUBES</RequestType>
  <Restrictions>
    <RestrictionsList>
      <CATALOG_NAME>Adventure Works DW
</CATALOG_NAME>
    </RestrictionsList>
  </Restrictions>
  <Properties>
    <PropertiesList>
      <Catalog> Adventure Works DW
```

```
</catalog>

        <Format> Tabular </Format>

    </PropertiesList>

</Properties>

</Discover>
```

MDX – Multidimensional Expressions

MDX [IME 2007] é a linguagem mais usada para estipular consultas OLAP sobre dados multidimensionais. Ela pode ser utilizada em conjunto com XMLA, retornando os resultados neste formato. A MDX possui características similares ao SQL, sendo rica e poderosa para a manipulação de dados e utilizando expressões compostas por identificadores, valores, funções e operadores. Permite estipular e interpretar consultas, formatar os resultados das consultas, bem como efetuar tarefas administrativas sobre um cubo dimensional. Uma possível consulta MDX sobre o cubo apresentado na Figura 5 é apresentada no Quadro 4:

Quadro 4 - Consulta MDX

```
SELECT {[Measures].[Num Candidatos]}
ON
COLUMNS,
{[Sexo].[Feminino],[Sexo].[Masculino]}
ON ROWS FROM VestibularUFSC
WHERE ([Local].[Rio Grande do Sul])
```


2.4 Sistemas de Informação Geográfica

Um Sistema de Informação Geográfica (GIS – Geographic Information System) [GIS 2007; WIG 2007] é um tipo de sistema de informação que permite capturar, analisar e gerir dados que têm uma perspectiva geográfica. GIS é uma ferramenta com muita utilidade nos dias de hoje, pois a sua informação pode ser usada para os mais diversos propósitos, tais como investigação científica, gestão de recursos, planeamento urbano, previsão e tratamento de catástrofes naturais.

2.4.1 Técnicas de GIS

Coleta de Dados

As tecnologias de GIS mais modernas usam informação digital, assim existem diferentes métodos de criação de dados. A técnica mais comum de criação de dados é a digitalização, por exemplo, copias de mapas são digitalizados e passados para computador através de um CAD (*Computer Aided Design*).

Representação de dados:

Dados de GIS representam objetos reais (ruas, terras, elevações). Os objetos são divididos em 2 abstrações: Objetos discretos (ex.: uma casa) ou objetos contínuos (ex.:chuva). Existem duas maneiras de representar estes objetos por Matriz e por Vector.

Matriz

Os dados pela forma matricial (ou *raster*) consistem num conjunto de linhas e colunas de células em que dentro de cada célula está guardado um único valor. O valor em cada uma das células vai representar o respectivo objeto. Esse valor pode ser uma variável discreta, que nesse caso pode representar um pedaço de terra, ou então pode ser uma variável contínua, que nesse caso poderia representar a chuva. Imagens *raster*

podem ser armazenadas em diferentes tipos como JPEG ou TIF.

Através da análise da Figura 6 é possível constatar que cada local da matriz corresponde a uma variável.

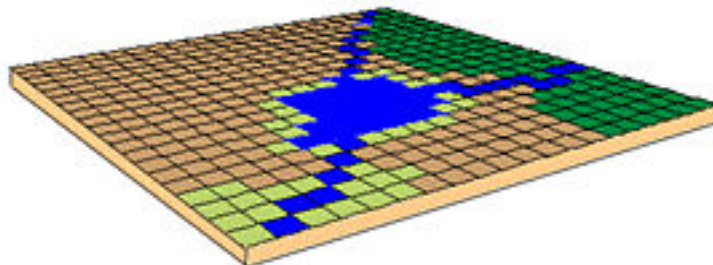


Figura 6 - Representação matricial

(<http://www.gis.unbc.ca/courses/geog300/lectures/lect2/rasterv.gif>, acessado em 20/03/2008)

Vetor

Dados representados por vetores usam a geometria para representar o espaço geográfico, como, por exemplo, pontos, linhas e polígonos para representar objetos. Estes dados também podem representar objetos contínuos ou discretos.



Figura 7 - Representação vectorial

(<http://www.gis.unbc.ca/courses/geog300/lectures/lect2/rasterv.gif>, acessado em 20/03/2008)

Vantagens e Desvantagens

Existem uma série de vantagens e desvantagens por usar o método matricial ou de vetor para representar a realidade. Nos dados por matriz é necessário ter um valor para todos os pontos na área que interessa representar, o que pode significar necessidade de um maior espaço para armazenamento, apesar de ser de mais fácil implementação do que por vetor. Com vetor, a imagem fica com uma maior noção de profundidade e com uma maior precisão do que com imagem *raster*.

2.4.2 Componentes da arquitetura de um GIS

Cada Sistema de Informação Geográfica possui uma arquitetura específica, no entanto existem certos componentes e funcionalidades comuns em todos e que fazem com que existam muitas semelhanças nas arquiteturas.

Para que um sistema de informações geográficas (GIS) utilize ao máximo os seus recursos, necessita ter na sua composição diversos componentes que, integrados, agregam grande utilidade a estes sistemas, sendo os principais:

- a) interface com usuário;
- b) entrada e integração de dados;
- c) consulta e análise espacial;
- d) visualização e plotagem;
- e) armazenamento e recuperação de dados sob a forma de um banco de

dados geográficos.

3 Análise de ferramentas

O objetivo desta pesquisa não é a análise das ferramentas disponíveis para implementação de soluções integradas OLAP/GIS, no entanto as dificuldades encontradas em certos componentes desenvolvidos fazem com que tenham sido estudadas diversas alternativas para esta mesma problemática. Desta forma foi considerado ser de extrema importância para futuros estudos desta temática a explicitação do estudo das ferramentas efetuado.

3.1 *Microsoft Visual Studio*

O Microsoft Visual Studio é o principal ambiente de desenvolvimento integrado da Microsoft (IDE). Este IDE permite o desenvolvimento das mais variadas aplicações, que vão desde consola e gráficas até páginas Web, WebServices e ferramentas de BI.

Para facilitar o desenvolvimento rápido e eficaz do programador, o Visual Studio possui um editor de código que analisa o código sem ser necessária a compilação e arranja o código de forma a ser mais facilmente lido, com indentação automática e sugestões que facilitam a escrita. A suíte completa possui várias linguagens de programação pré-definidas como o C/C++; VB:NET e o C#.

O facto desta ferramenta ser um ambiente de desenvolvimento integrado, foi possível agregar todos os componentes da arquitetura num só local, o que impediu problemas de integração e facilitou a implementação da comunicação entre os componentes.

3.2 Servidor OLAP

O Servidor OLAP é o servidor responsável por processar as consultas OLAP que o usuário desejou sobre o sistema gerenciador de banco de dados – SGBD. Este possui uma interface própria para consultas de dados multidimensionais, retornando depois o resultado da consulta. Neste trabalho foram encontrados dois caminhos: através do servidor OLAP da Microsoft (Analysis Services) ou através de um servidor OLAP Open Source (Mondrian). O caminho escolhido foi o do Analysis Services da Microsoft, baseado na facilidade de utilização desta ferramenta, que é mais intuitiva e fácil de usar que a outra opção. Em baixo se detalham as duas ferramentas.

3.2.1 Microsoft SQL Server 2005 Analysis Services

O SQL Server 2005 é uma plataforma abrangente de banco de dados, que fornece recursos de gerenciamento de dados de classe empresarial com ferramentas de BI (Business Intelligence) integradas, das quais, o Analysis Services faz parte [Microsoft Brasil, 2008b].

O Microsoft Analysis Services é uma ferramenta de Business Intelligence que oferece um conjunto de serviços usados para gerenciar dados num Data Warehouse ou Data Mart. O Analysis Services organiza dados de um Data Warehouse em cubos, utilizando o conceito de visão multidimensional para permitir a geração de relatórios sofisticados e consultas complexas, sendo capaz de tratar informações em grandes quantidades de dimensões.

Esta ferramenta inclui assistentes (wizards) de projeto e implementação dos cubos que tornam a sua manipulação e consulta de dados muito mais fácil do que manualmente. A conexão é feita via ODBC e permite fontes de dados em diversos formatos.

O Microsoft Analysis Services é um conjunto de serviços usados para gerenciar dados em um Data Warehouse ou Data Mart. O Analysis Services organiza dados de um Data

Warehouse em cubos utilizando o conceito de visão multidimensionais para permitir a geração de relatórios sofisticados e consultas complexas, sendo capaz de tratar informações em grandes quantidades de dimensões.

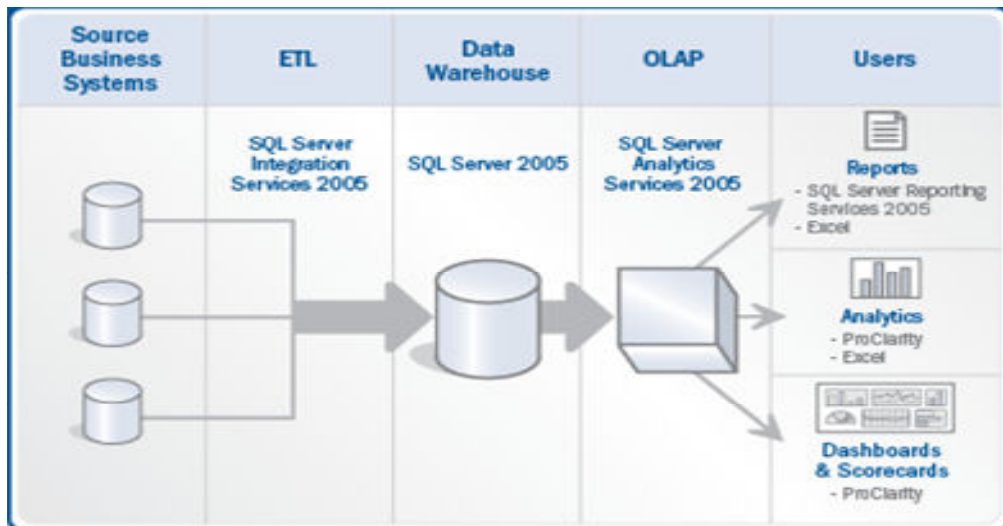


Figura 8 - Interação das ferramentas nas diferentes camadas de BI [NORTHRIDGE SYSTEMS, 2008]

A Figura 8 mostra que as grandes soluções de BI são compostas por diversas camadas cada uma delas com um papel e ferramentas bem específicas.

Na Figura 9 é possível identificar onde é que cada uma dessas camadas se encontram na nossa arquitetura.

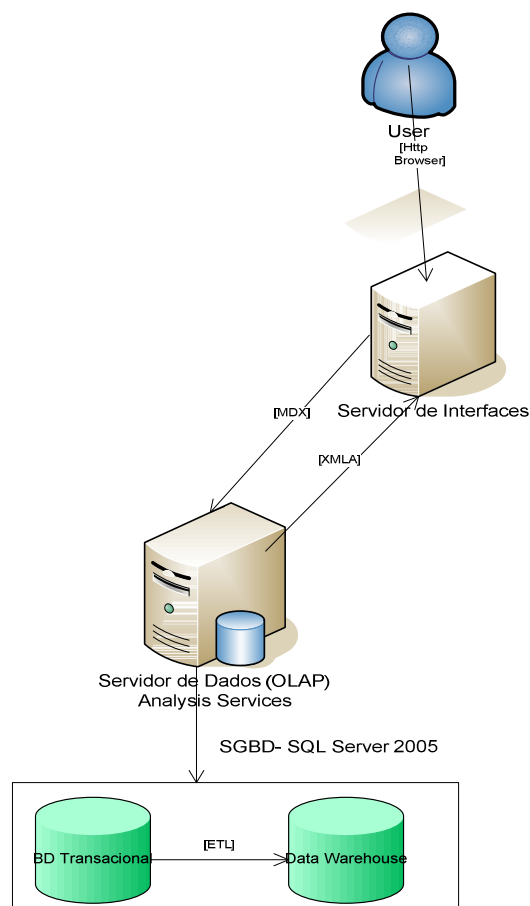


Figura 9 – Lado do DW e localização do Analysis Services

O Microsoft Analysis Services permite efetuar análises OLAP via Web sobre um cubo dimensional implementado neste através do objeto de comunicação ADOMD.

Para utilizar este objeto de ligação no editor de código é preciso importar uma biblioteca através do seguinte comando:

```
using Microsoft.AnalysisServices.AdomdClient;
```

Outro passo importante para estabelecer a comunicação com o servidor de dados dimensionais via Web Service é o arquivo XML WebConfig, pelo qual é detalhada a forma de comunicação no *.NET*.

No Quadro 5 e Quadro 6 seguem os exemplos do uso do objeto de comunicação ADOMD e de como o arquivo WebConfig foi construído:

Quadro 5 - Objeto de comunicação ADOMD

```
AdomdDataAdapter data_adapter;
AdomdConnection conn1;
conn1 = new AdomdConnection("Provider=SQLNCLI.1;Data
Source=localhost;Integrated Security=SSPI;Initial Catalog=Analysis Services
UFSC DW");

data_adapter = new AdomdDataAdapter("SELECT NON EMPTY { [Measures].[Num
Candidatos] } ON COLUMNS, NON EMPTY { ((DISTINCT([Candidato].[Estado
Endereco].[Estado Endereco])) * [Candidato].[Sexo].[ "+sexo+" ] *
[Candidato].[Raca].[ "+raca+" ] ) } DIMENSION PROPERTIES MEMBER_CAPTION,
MEMBER_UNIQUE_NAME ON ROWS FROM [UFSC DW]", conn1);
```

O funcionamento deste objeto de comunicação inicia-se através da criação de um objeto *DataAdapter*, que é o objeto responsável por guardar a informação proveniente do banco de dados. Posteriormente, é criado um objeto de conexão que, no Quadro 5 chama-se *conn1*, e coloca-se a respectiva *string* com o endereço que permite estabelecer a ligação ao servidor OLAP, que neste caso se chama “*AnalysisServices UFSC DW*”. Por fim, é carregado o *dataAdapter* com a query que se pretende visualizar. Neste exemplo o usuário pretende visualizar o estado endereço, o sexo e a raça dos candidatos da UFSC.

Quadro 6 - Arquivo WebConfig.xml

```
<connectionStrings>
  <add name="UFSC DWConnectionString" connectionString="Data
Source=localhost;Initial Catalog=&quot;UFSC DW&quot;;Integrated
Security=True" providerName="System.Data.SqlClient"/>
</connectionStrings>

<assemblies>
  <add assembly="Microsoft.AnalysisServices.AdomdClient,
Version=9.0.242.0, Culture=neutral, PublicKeyToken=89845DCD8080CC91" />
</assemblies>
```

O arquivo do Quadro 6 é muito importante pois é aqui que são colocadas as referências aos objetos e as ferramentas utilizadas no projeto. Este documento é criado de forma quase automática, pois possui um assistente que facilita em muito a tarefa do usuário que só tem que dizer quais as ferramentas que deseja utilizar no seu projeto. O caminho para essas mesmas ferramentas é traçado no documento *WebConfig.xml* automaticamente [Microsoft Brasil, 2008b], como o Quadro 6 demonstra.

Esta ferramenta inclui wizards para projeto e implementação dos cubos e conexão via ODBC com fontes de dados em diversos formatos, para apoiar o processo de extração, transformação e carga de dados. Após a criação do cubo é feito o *deployment* deste de modo a permitir efetuar as consultas desejadas.

Para criar um cubo existem diversos passos a tomar:

1º Passo – Escolher a Data Source do cubo Definir quais são os dados que irão constituir o cubo e estabelecer a ligação a conexão via ODBC.

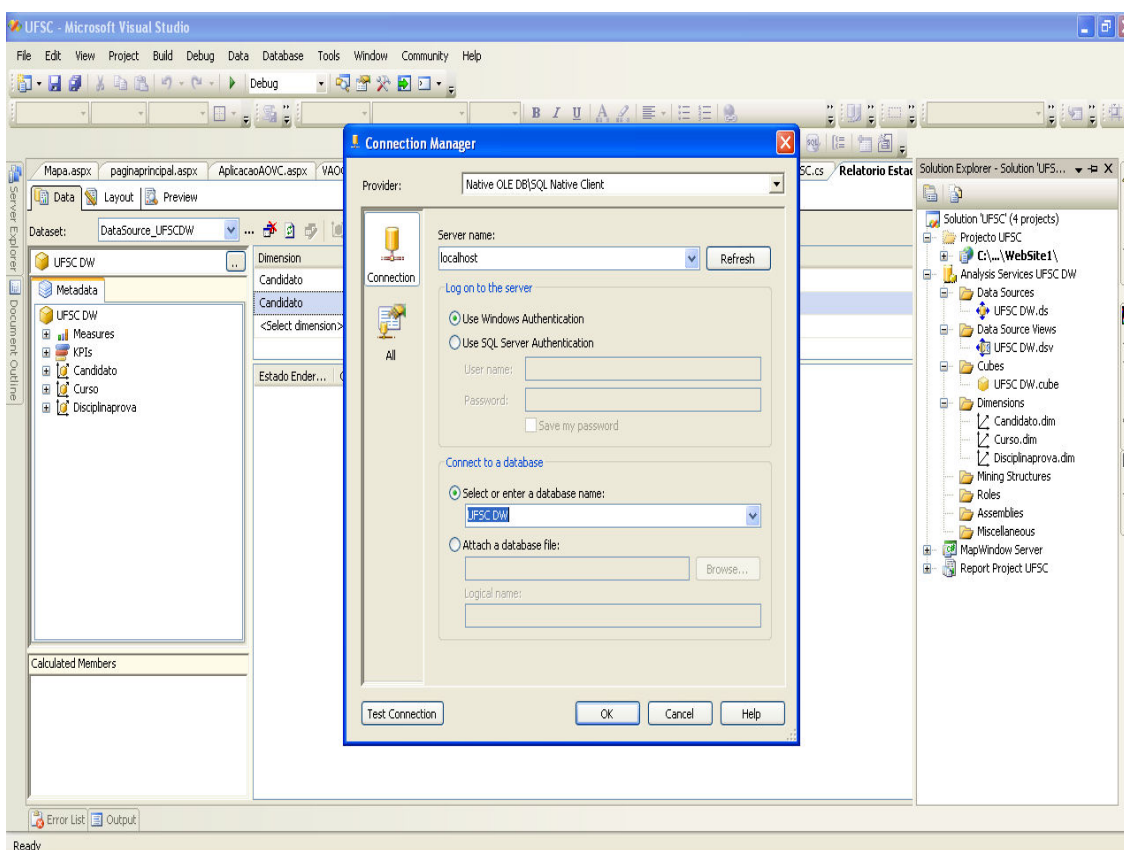


Figura 10 - Conexão via ODBC

Este é um exemplo de um dos wizards desta ferramenta, que se destaca pela facilidade de uso. Neste caso específico basta apenas escolher a localização e nome do servidor, que na Figura 10 está no *localhost* sob o nome de *UFSC DW* servidor que no nosso caso pusemos a nossa própria máquina ou seja o *localhost* e a base de dados respectiva que neste caso era a *UFSC DW*, carregávamos no OK e a nossa Data Source estava criada.

2º Passo – Selecionar a Data Source View, ou seja escolher quais as tabelas que são importadas, que neste caso são as tabelas de fato e as tabelas de dimensões. Mais uma vez é uma tarefa facilitada graças aos wizards que esta ferramenta possui. Na Figura 11 encontra-se o *wizard* deste passo.

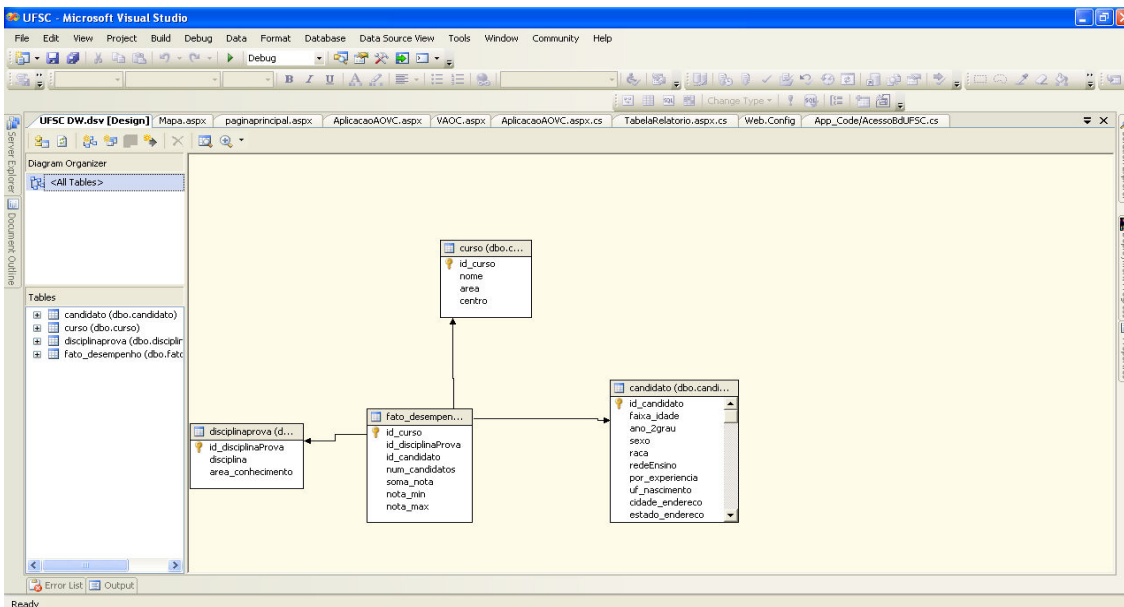


Figura 11 - Criação do StarScheme no Analysis Services

Neste trabalho é utilizado o modelo StarScheme. O cubo do caso de estudo é simples e é constituído por uma tabela de Facto que é a tabela que se encontra no meio, e por três tabelas de dimensões, as tabelas que rodeiam.

3º Passo – Criar o cubo. Na Figura 12 está o *Wizard* que auxilia o usuário a criar o cubo, com os dados, e o modelo dimensional que são necessários. Novamente, o *wizard* facilita a criação do cubo ao usuário.

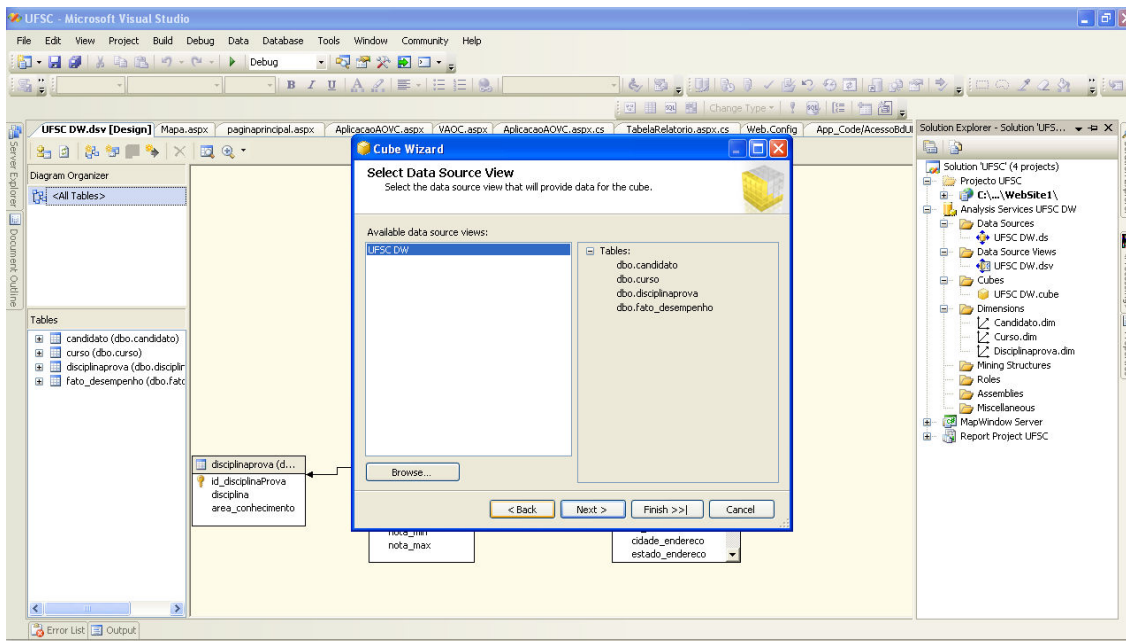


Figura 12 - Wizard de criação do Cubo

3.2.2 Pentaho Analysis Services: Mondrian

Mondrian é um servidor OLAP Open Source que foi desenvolvido em *Java*. Este servidor suporta a linguagem multidimensional MDX e permite ao usuário interagir e manipular grandes bancos de dados. Permite também efetuar cálculos avançados através de expressões MDX, estabelecer indicadores de desempenho de modo a conseguir melhores respostas. O Mondrian utiliza JDBC para conexões com o SGBD (Sistema de Gerenciamento de Banco de Dados) utilizado para manter os dados do Data Warehouse [MACDONALD,2007].

Este servidor não foi alvo de estudo aprofundado, pois o Analysis Services da Microsoft foi à partida uma ferramenta que satisfaz todas as necessidades para concluir este projeto.

O Quadro 7 apresenta uma tabela comparativa dos dois servidores OLAP analisados.

Quadro 7 - Comparação de servidores OLAP

Tecnologia	Vantagens	Desvantagens
MS Analysis Services	Maior facilidade de utilização, mais perceptível	Produto pago
Mondrian	Open Source, maior facilidade de personalização	Pouco intuitivo, maior dificuldade de manipulação

3.3 Interface OLAP

Uma das principais dificuldades encontradas neste trabalho de pesquisa foi a criação de uma Interface OLAP que permitisse ao usuário visualizar o resultado da sua pesquisa. Várias soluções foram discutidas e uma vez mais foi selecionada uma solução da família Microsoft, o “Reporting Services”.

3.3.1 Desenvolvimento de raiz

A primeira opção foi a de desenvolver a própria interface. O objetivo não foi concluído com o êxito desejado, pois, embora o resultado da pesquisa do usuário fosse mostrado na tela, falhou no que trata à sua apresentação estética. A apresentação dos dados era algo confusa e complicada e não era possível relacionar todos os dados entre si, perdendo assim a grande vantagem dos dados multidimensionais.

Implementação

Nesta implementação feita de raiz, foi necessário seguir um conjunto de passos que são descritos a seguir:

1º Passo – Foi estabelecida uma ligação entre o cubo dimensional e o Web site, implementando uma interface OLAP básica através do objeto de ligação ADOMD. Este objeto permite retornar todos os resultados das consultas efetuadas sobre cubo dimensional para posterior apresentação na tela.

2º Passo – Com os dados do resultado da consulta efetuado pelo usuário, é necessário organizar e estruturar os dados. A primeira falha desta implementação iniciou-se aqui, pois não foi possível colocar numa matriz a relação dos dados entre si, ou seja só foi possível colocar cada um deles individualmente, perdendo assim a multidimensionalidade. A impossibilidade de criar uma matriz a partir de uma consulta OLAP deveu-se ao facto de não haver recursos suficientes no tempo disponível para uma implementação deste tipo. A alternativa foi colocar cada um desses dados em tabelas diferentes.

Quadro 8 - Organização dos dados da tabela

```
BoundField field1 = new BoundField();
BoundField field2 = new BoundField();
BoundField field3 = new BoundField();
BoundField field4 = new BoundField();

field1.DataField = "[" + dimensao1 + "].[" + atributo_dimensao1 + "].[" +
atributo_dimensao1 + "].[MEMBER_CAPTION]";
field1.HeaderText = atributo_dimensao1;
GridView1.Columns.Add(field1);

field2.DataField = "[" + dimensao2 + "].[" + atributo_dimensao2 + "].[" +
atributo_dimensao2 + "].[MEMBER_CAPTION]";
field2.HeaderText = atributo_dimensao2;
GridView1.Columns.Add(field2);

field3.DataField = "[" + dimensao3 + "].[" + atributo_dimensao3 + "].[" +
atributo_dimensao3 + "].[MEMBER_CAPTION]";
field3.HeaderText = atributo_dimensao3;
GridView1.Columns.Add(field3);
```

Os dados na tabela foram organizados através da criação de uma GridView, que é o objeto responsável por mostrar dos dados numa tabela. Os campos que fazem parte da

consulta foram criados em BoundField, que junta os dados da consulta e que, posteriormente, foram inseridos com o resultado da consulta. Finalmente, foram adicionadas as colunas da GridView.

3º Passo – O último passo foi colocar a solução disponível numa página web na Internet.

Na Figura 13, encontra-se um exemplo da antiga aplicação, que mostra em três tabelas os resultados de uma pesquisa feita ao Estado Endereço, à Faixa de Idade e ao Sexo.

Estado	Endereço	Num. Candidatos
		636912
Acre		48
Alagoas		168
Amapá		24
Amazonas		48
Bahia		648
Ceará		72
Distrito Federal		4800
Espírito Santo		360
Goiás		2496
Maranhão		24
Matto Grosso		3096
Matto Grosso do Sul		4512
Minas Gerais		3408
Para		120
Paraíba		24
Paraná		34416
Pernambuco		48
Rio de Janeiro		1104
Rio Grande do Sul		21504
Rondonia		384
Santa Catarina		500256
Sao Paulo		58968
Sergipe		264
Tocantins		120

Faixa Idade	Num. Candidatos
	636912
18 a 20	274368
21 a 23	65688
24 a 26	28032
Até 17	230760
Mais de 26	38064

Sexo	Num. Candidatos
	636912
F	329616
M	307296

Figura 13 - Resultado de uma consulta

3.3.2 Microsoft SQL Reporting Services

O SQL Server Reporting Services (SSRS) é uma ferramenta criada pela Microsoft e faz parte do framework de Business Intelligence da Microsoft. É baseada em servidor e permite a criação, o gerenciamento e a entrega de relatórios tradicionais, baseados em papel ou em ambiente Web, com alta interatividade.

A sua principal função é realizar relatórios sobre os dados existentes num determinado banco de dados [MICROSOFT, 2008]. Tanto podem ser definidos relatórios estáticos, em que o usuário apenas consulta o relatório previamente definido, como podem ser definidos relatórios dinâmicos, em que o usuário consiga criar de forma dinâmica os seus próprios relatórios.

Esta ferramenta combina as capacidades de gerenciamento de dados do SQL Server e o Microsoft Windows Server com aplicações que fornecem informações em tempo real para suportar as operações diárias e auxiliar na percepção de negócios para tomada de decisões mais precisas [MICROSOFT BRASIL, 2008a].

Esta ferramenta é muito poderosa, pois responde às necessidades atuais do mercado no que respeita à fácil percepção de grandes quantidades de dados, pois apresenta os dados de uma forma simples e intuitiva.

Os diferentes tipos de relatório:

O SQL Server Reporting Services combina uma plataforma única e completa com uma arquitetura escalável e extensível para satisfazer muitas das necessidades de relatórios, incluindo [MICROSOFT, 2008]:

Relatórios corporativos. As empresas podem usar o Reporting Services para suas aplicações de BI ou relatórios operacionais. Com o Reporting Services, a equipe de TI

pode desenhar vários relatórios e implantá-los tanto de forma individual como para toda a empresa.

Relatórios Ad-hoc. O SQL Server 2005 Reporting Services inclui o Report Builder, uma nova ferramenta ad-hoc de relatórios que permite aos usuários de negócios criarem seus próprios relatórios e explorarem os dados corporativos. O Report Builder incorpora um modelo de consulta que permite aos usuários criarem relatórios sem precisar de um entendimento técnico profundo das fontes de dados subjacentes.

Relatórios embutidos. As organizações podem acessar relatórios ad-hoc ou pré-definidos a partir de aplicações de terceiros que utilizam o Reporting Services e então usar estes relatórios como estão, personalizá-los ou criar novos para necessidades específicas de negócio.

Relatórios Web para parceiros e clientes. As organizações podem implantar relatórios baseados na Web para entregar informações para clientes ou parceiros através de Extranets na Internet. O Reporting Services isola os relatórios de clientes da complexidade de fontes de dados subjacentes, enquanto fornece personalização e maior interatividade.

Características do Microsoft SQL Server Reporting Services [MICROSOFT, 2008]:

Report Builder: O Report Builder é uma ferramenta de consulta de relatórios ad-hoc destinada ao usuário final, pois possibilita relatórios sofisticados de auto-atendimento e cenários de exploração de dados que permitem aos usuários criarem e modificarem seus próprios relatórios.

Model Designer: O Model Designer permite definir, editar e publicar modelos de relatórios para serem usados futuramente com o Report Builder.

Report Designer: O Report Designer é um ambiente de desenvolvimento baseado no

Visual Studio que possibilita a definição dos tipos mais complexos de relatórios.

Report Manager: O Report Manager é uma ferramenta baseada na Web que fornece um único local para acessar e gerenciar todos os relatórios.

Report Viewer: O SQL Server 2005 Reporting Services fornece aos usuários múltiplas opções para visualização de relatórios de forma interativa. [Microsoft Brasil, 2008b]

A ferramenta Implementada:

Para implementar os relatórios necessários no protótipo, foi necessário seguir um conjunto de passos, sendo que muitos desses tiveram a ajuda de assistentes que existem nestas ferramentas:

1º Passo - Definir o Data Source – Neste primeiro passo é necessário definir a origem dos dados. Neste caso específico provém de um cubo chamado *Analysis Services UFSC DW*. Colocou-se a seguinte string de ligação “Data Source=localhost;Initial Catalog="Analysis Services UFSC DW"” Nesta String definiu-se que o servidor utilizado é local, e que os dados provêm de uma fonte chamada Analysis Services UFSC DW que representa o DW do caso de estudo.

2º Passo – É necessário definir quais são os dados que se pretende visualizar, ou seja, para isso é necessário criar uma consulta que vá ao cubo fazer um pedido, que no caso do *Microsoft SQL Reporting Services*, existem 2 maneiras possíveis de fazer isso. A primeira, mais fácil, é utilizar o *Query Builder*, que é um wizard que permite utilizar drag and drop com os dados que se deseja visualizar e a própria aplicação cria a consulta pelo usuário. A segunda maneira é mais complexa, pois o usuário faz a consulta manualmente, ou seja, é ele a escrever o próprio comando. Neste caso específico foi definida uma

consulta para obter o UF Nascimento, a Area e a Raça, conforme o Quadro 9 o demonstra:

Quadro 9 - Exemplo de uma consulta MDX

```
SELECT {[Measures].[Num Candidatos]} ON  
COLUMNS, {[Sexo].[Feminino],[Sexo].[Masculino]}  
ON ROWS FROM VestibularUFSC  
WHERE ([Date].[Year].[2006])
```

3º Passo - Neste passo é necessário estruturar os dados, ou seja definir como se deseja visualizar os dados e organizá-los. Existem 2 maneiras de visualizar os dados: em forma tabular, como exemplificado na Figura 13 ou em forma matricial, como exemplificado na Figura 14, sendo que neste caso a opção foi a da visualização matricial. Após a escolha de como se deseja visualizar os dados é necessário organizar os dados na matriz, escolher em que coluna e em que linhas ficam cada um dos campos da consulta efetuada no passo anterior.

4º Passo- Por fim faz-se a implantação do relatório de modo a que este seja possível ser visível na Web, sendo que para isso é necessário colocar o relatório num servidor. O Microsoft SQL Reporting Services cria automaticamente uma pasta virtual chamada Report Servers que permite a disponibilização imediata na web.

Após estes 4 passos o relatório está concluído, na Figura 14 está um exemplo de um relatório enquadrado no caso de estudo.

1 of 2 100% Find | Next Se

UFNascimentoAreaRaca

		Ciências Biológicas	Ciências Exatas	Ciências Sociais
Acre	BRANCA	36	36	60
	PARDA	12	12	12
Alagoas	AMARELA	12		
	BRANCA	12	60	12
	PARDA			36
Amapá	BRANCA			12
	PARDA			12
	PRETA		12	
Amazonas	AMARELA			12
	BRANCA	36	36	12
	INDÍGENA		12	
	PARDA	12		24
	PRETA		12	12
Bahia	BRANCA	84	264	72
	PARDA	36	96	132
	PRETA	12	12	72
Ceará	AMARELA			12
	BRANCA	48		48
	PARDA		12	48

Figura 14 - Exemplo de relatório usando o Reporting Services

3.4 Ferramentas GIS

A prioridade foi o estudo de ferramentas proprietárias da *Microsoft* e, por isso, este ponto inicia-se com o relato das conclusões sobre o software desta. Como não foi possível fazer o desenvolvimento com a ferramenta estudada, foi necessário procurar outras, de forma a prosseguir este estudo com sucesso.

3.4.1 Microsoft MapPoint 2006 North America

Esta versão da aplicação destina-se, principalmente, para usuários da América do Norte, pois as suas totais capacidades estão somente disponíveis nessa área Geográfica.

Além desta hipótese, existiam outras duas dentro da família de produtos MapPoint [MAPPOINT, 2008]:

- Microsoft MapPoint 2006 Europe Edition – Mesmas funções que a aplicação estudada, mas focada fundamentalmente para o território europeu
- Microsoft MapPoint WebService – serviço Web suportado pela Microsoft que possui informação detalhada sobre o Brasil. Necessita de licença para conectar ao servidor e, conseqüentemente, ao serviço.

A opção pelo estudo da aplicação indicada deveu-se ao facto de esta ser a única ferramenta da família MapPoint com licença disponível.

Esta foi uma opção estudada a fundo pois foi a ferramenta em que inicialmente definida para criar a visualização geográfica. As funcionalidades da aplicação eram promissoras, pois a existência de um add-in que realizava queries OLAP directamente ao Data Warehouse e outro que permitia a importação de arquivos com extensão *.shp* ajudariam na execução do que foi proposto.

O API disponibilizada pelos dois add-ins acima mencionados caracteriza-se pela falta de documentação e poucas funcionalidades facultadas ao programador, aliadas à escassez de tempo impossibilitaram a continuidade do projeto na ferramenta MapPoint, pelo que foi necessário procurar novas abordagens à solução do problema proposto.

Qualquer uma das soluções acima descritas possui licença comercial, pelo que é necessário efetuar pagamento para poder utilizar este software sem ser por tempo limitado.

Após o estudo efetuado nesta ferramenta, fica clara a ideia que esta é bastante apropriada para problemas de criação de rotas, geração de gráficos distribuídos geograficamente e funcionalidades de GPS dentro da área geográfica a que diz respeito.

3.4.2 MapWindow GIS

Esta é uma solução de GIS open source e gratuita que se caracteriza por ser extensível, pois permite que qualquer pessoa consiga adicionar funcionalidades à aplicação [MAPWINDOW, 2007].

Esta solução apresenta três produtos distintos:

- MapWindow Application – aplicação desktop que permite a visualização espacial de dados.
- ActiveX control – este componente é um objeto programável que pode ser embutido numa janela applicativa. Possui um API que permite o acesso a dados de polígonos, tabelas e imagens. Pelas características atrás descritas este componente torna-se bastante poderoso pois permite a criação de um GIS à medida das necessidades do usuário.
- Plug-ins – é possível acrescentar funcionalidades à aplicação desktop MapWindow. A sua principal vantagem é possuir um controlo activeX programável que pode ser facilmente embutido em qualquer aplicação, a par das capacidades que o seu API permite ao programador.
- MapWindow Server 5 – Ferramenta muito recente destinada a aplicações em ambiente Web. Devido à sua fase inicial de desenvolvimento necessita ainda de muita documentação e melhorias para que se possa iniciar uma implementação consistente e fiável. A ser estudado com profundidade em futuras implementações de GIS baseados na Web.

Outra característica bastante positiva para o desenvolvimento deste projeto é o fato de ser gratuito e permitir um suporte bastante interessante para iniciantes.

A escolha recaiu sobre esta ferramenta pois as características do componente activeX desta aplicação permitem a manipulação dos dados conforme os requisitos do projeto.

3.4.3 ArcGIS Explorer

Esta aplicação é um visualizador de informação geográfica muito eficaz, pois permite a visualização dos dados de diversas formas, devido às muitas funcionalidades que possui [ESRI, 2007].

Além das mesmas funcionalidades dos GIS comuns, o ArcGIS Explorer permite juntar dados locais com dados recolhidos existentes em servidores da ESRI e de parceiros e analisar geograficamente dados, quer estáticos quer através de tarefas (procuras de locais, modelação, análise de visibilidade). Permite também receber dados provenientes de Web Services.

Ao nível de aplicação desktop, este é um visualizador GIS que se pode adaptar às necessidades do usuário, pois permite a personalização da aplicação.

Possui um SDK disponível, quer permite a criação de tarefas existentes e novas para a aplicação, no entanto este é complexo e não possui foruns de ajuda e documentos suficientes para que esta hipótese fosse estudada mais seriamente.

3.4.4 AvisMap GIS Engine

O AvisMap GIS Engine [AVISMAP GIS, 2007] é uma plataforma de desenvolvimento de GIS que se baseia em controlos ActiveX. As aplicações criadas nesta plataforma permitem que as suas distribuições possam ser instaladas nas máquinas dos clientes sem que estes possuam o GIS Engine.

Esta plataforma possui muitas capacidades, e deve ser analisada muito seriamente se se pretender implementar um GIS mais complexo e que requeira funcionalidades abaixo descritas:

- Visualização e edição de mapas complexos – com manuseio de elementos 3D, projeção dinâmica, conversor de dados vetoriais para matriciais e vice-versa, entre

outras funcionalidades;

- Criação de mapas temáticos – organizados por valores, por segmentos, por símbolos, pelos quais é possível visualizar gráficos 3D para uma melhor percepção dos dados envolvidos;
- Análise e processamento topológico – permite que seja feita a remoção de dados redundantes, nomeadamente vértices, linhas e polígonos, assim como juntar nós adjacentes e construir polígonos;
- Funções de análise espaciais – modelação 3D, operações algébricas matriciais, de análise hidrológicas, entre outras;

Esta ferramenta possui muitas funcionalidades, que vão auxiliar o desenvolvimento de um GIS mais complexo do que aquele que é necessário para a concretização deste projeto.

O principal aspecto contra é o fato de ser necessário efetuar pagamento para possuir uma licença de utilização do software, pelo que neste contexto, torna muito difícil a realização deste projeto sobre esta ferramenta.

3.4.5 Quantum GIS

O Quantum GIS é um GIS de código aberto extensível e flexível, o que significa que se podem criar plug-ins para adicionar à aplicação e também possibilita a customização desta através da programação de novas ou tarefas mais especializadas [QUANTUM GIS, 2007].

Esta aplicação, de base não possui nenhuma capacidade diferenciadora dos comuns GIS, no entanto consegue ser bastante abrangente pois suporta inúmeros tipos de formato de dados para análise geográfica.

O Quantum GIS é mantido e atualizado por programadores voluntários de todo o

mundo, fazendo com que o suporte e mecanismos de ajuda à aplicação sejam melhores que muitas aplicações livres existentes.

Apesar das funcionalidades base não se distinguirem dos GIS de software livre comuns, a sua extensibilidade e plug-ins já disponíveis faz com que esta se torne numa ferramenta a ter em conta no desenvolvimento de um GIS.

3.4.6 Comparativo

Na Figura 15 está um comparativo com as ferramentas GIS estudadas. O principal critério, depois de falhados os experimentos com o Microsoft MapPoint 2006, foi o da facilidade de implementação e da integração com padrões de ficheiros de formatos geográficos.

O MapWindow destacou-se dos outros GIS estudados pelos critérios definidos e também pelo fato de estar em fase de expansão e assim abrir possibilidades de melhoria neste software.

GIS	Linguagens	Ajuda	Facilidade de implementação	Formatos de ficheiros	Plataforma	Preço	Vantagens	Desvantagens
MapPoint 2006 North America	C#, C++, Delphi, Pearl, Python, VB.NET, VBA	Forums com muita interação entre utilizadores; Websites de ajuda; tutoriais completos e variados	Média	MS Excel, MS Access, Ficheiros de texto (.txt, .csv)	Windows	\$299 USD	Possui add-in que permite o mapeamento de query OLAP directamente	Ferramenta paga, pouca integração com formatos de ficheiros mundialmente aceites
MapWindow	C#, C++, Delphi, Pearl, Python, VB.NET, VBA	Forums com nível de ajuda básico, tutorial para principiantes, código disponibilizado por outros utilizadores, livros publicados	Elevada	ESRI Shapefile, ASCII grid, formatos de imagem (JPEG, BMP)	Windows	Gratuito	Encontra-se em franca expansão e é uma ferramenta que concorre ao nível de capacidades com GIS pagos	
ArcGIS explorer	C# e VB.NET	Documentação para principiantes	Baixa	ESRI Shapefile e outros padrões da mesma empresa	Windows	Gratuito	Pode ser agregado com outras ferramentas da ESRI	Pouca ajuda, ferramenta limitada
AvisMap Engine	C#, C++, Delphi, Pearl, Python, VB.NET, VBA	Documentos online para principiantes e programadores	Média	ESRI Shapefile, MapInfo, KML, GML, TIFF/GeoTIFF, formatos de imagem (JPEG, BMP)	Windows	\$199 USD	Ferramenta muito completa, aceita a maioria dos formatos standart de dados geoespaciais	Difícil encontrar comunidade de utilizadores desta ferramenta
Quantum GIS	C/C++	Documentos de ajuda para programadores, documentação da API, blog, forum, IRC e contactos de programadores para ajuda	Elevada	ESRI Shapefile, MapInfo, Arc/Info binary converage, SDTS, Arc/info Binary Grid, GRASS Rasters	Windows XP, Linux, Unix, MAC OS	Gratuito	Comunidade de utilizadores em crescimento, open source, muita ajuda existente, multiplataforma	Linguagens de programação restritas

Figura 15 - Quadro comparativo das ferramentas GIS estudadas

4 Arquitetura proposta

Esta seção apresenta a arquitetura do sistema VAOC (Visualização de uma Aplicação Olap que permite ver Cartograficamente os alunos da UFSC), voltado para a geração de mapas em uma interface OLAP sobre a Web, mediante a comunicação desta interface com um servidor de dados dimensionais (ou servidor OLAP) e um servidor de mapas, de maneira transparente para o usuário. A Figura 16 mostra a arquitetura do VAOC que é composta de três módulos principais:

Servidor de Dados Dimensionais: Responsável por processar as consultas OLAP estipuladas pelo cliente sobre o SGBD que mantém os dados dimensionais. O servidor de dados utilizado na nossa implementação foi o Microsoft Analysis Services conectado ao SQL Server 2005. A Microsoft SQL Reporting Services é a ferramenta sobre a qual foi implementada o componente da ferramenta VAOC para visualização dos relatórios por parte dos usuários.

Servidor de Mapas: Responsável por criar o mapa referente ao resultado de uma consulta OLAP, quando da solicitação pelo usuário, mediante o acionamento de um botão. Este servidor utiliza os dados dimensionais provenientes do Data Warehouse e as extensões geográficas mantidas em um GIS para construir o mapa. O protótipo realizado gera mapas no formato .bmp utilizando o MapWindow.

Servidor de Interfaces: Responsável por estabelecer a ligação entre o servidor de dados e o servidor de mapas. Representa o Web Service, pois estabelece a ligação entre duas aplicações diferentes. A interface deste servidor foi criada de raiz de modo a facilitar uma maior percepção e facilidade de uso por parte do usuário.

No Quadro 10 estão as ferramentas utilizadas nos dois servidores e qual o formato de dados que recebem e enviam para comunicar.

Quadro 10 - Características dos componentes

Servidor	Ferramenta	Formato dos Dados
Servidor de Dados	Microsoft Analysis Services	Pergunta: MDX Resposta: XMLA
Servidor de Mapas	MapWindow	Pergunta: XMLA Resposta: BMP

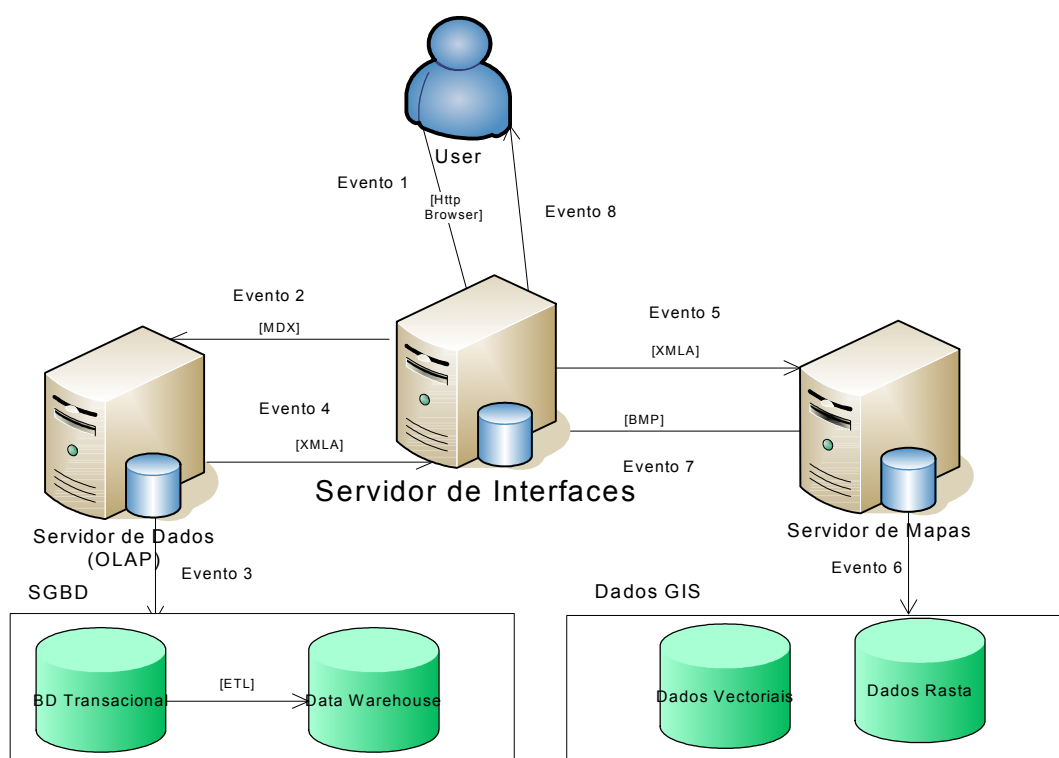


Figura 16 - Arquitetura do site VAOC

Para gerar um mapa, desde a solicitação do usuário até a visualização do mapa na tela, os eventos seguem esta seqüência:

1. O usuário interage com a interface OLAP, selecionando alguma visão do cubo de dados dimensional, através da aplicação de operadores com *drill-down* e *roll-up*;
2. O servidor de interface solicita ao servidor OLAP a execução dos comandos para a geração da visão de dados dimensionais, através de uma expressão MDX;
3. Ao receber a expressão MDX, o servidor OLAP realiza as consultas necessárias no DW, retornando os dados dessa visão do cubo em formato XMLA;
4. Ao receber o XMLA referente à visão de dados dimensionais, o servidor de interface gera a visualização tabular dos dados na tela do usuário;
5. Quando o usuário solicita a geração do mapa, para a visão tabular de dados dimensionais presente na sua tela, o servidor de interface transmite a requisição ao servidor de mapas, enviando o arquivo XMLA correspondente à visão, juntamente com o foco desejado;
6. O servidor de mapas, por sua vez, processa o arquivo XMLA, recolhendo as informações necessárias e convertendo-as em especificações para a geração do mapa;
7. Com os dados obtidos, o servidor de mapas os associa com suas respectivas extensões geográficas, que são buscadas no repositório do GIS, gerando o mapa, o qual é enviado ao servidor de interface;
8. Por fim, o servidor de interface apresenta o mapa gerado para o usuário.

5 Implementação

5.1 *MapWindow*

Conforme foi descrito no capítulo 3.4.2, a implementação foi feita através do ***MapWindow ActiveX component*** para a criação do GIS. Para a implementação da aplicação, foi necessário recorrer ao API desta solução disponível através de dois arquivos .DLL (Dynamic Link Library):

- AxInterop.MapWinGIS.dll – responsável pela manipulação do componente de ActiveX;
- Interop.MapWinGIS.dll – que possui os componentes necessários para a criação do GIS.

Os dados geográficos foram retirados do Instituto Brasileiro de Geografia e Estatística (IBGE) e encontram-se no formato ESRI Shapefile.

O desenvolvimento deste projeto tinha como referência ser elaborado para o ambiente Web, pelo que foi desenvolvida uma aplicação baseada no MapWindow que comunica com um Webservice e que possui duas grandes funcionalidades para responder ao desafio colocado:

- Gerar mapas
- Servir mapas

5.1.1 Gerar mapas

A aplicação por desenvolvida materializa-se numa janela applicativa que está situada na máquina em que o Web Service está presente e é nesta mesma janela que o componente ActiveX gera os mapas que são requisitados.

O pedido de geração do mapa escolhido pelo usuário chega à aplicação através de

um DataSet que contém uma tabela com a referência geográfica a ser representada e os valores respectivos de cada uma. Cada mapa pedido tem um arquivo shape associado a ele, que fica guardado numa pasta predefinida pela aplicação. Para que se saiba qual o arquivo shape que vai ser modificado, convencionou-se que a primeira coluna da primeira linha tem o nome hierarquicamente acima do mapa que se quer mostrar, ou seja:

- 1) Nome do país – mostra o país dividido pelos estados;
- 2) Nome do estado – mostra os municípios do estado escolhido.

Assim torna-se fácil saber qual o arquivo utilizado na geração do mapa, conforme o Quadro 11 o demonstra:

Quadro 11 - Processo de pesquisa do arquivo geográfico

```
String nome = tabelaResultadoDaQuery.Rows[0][0].ToString();
String[] arquivos = System.IO.Directory.GetFiles("C:\\Folder");

for (int i = 0; i < arquivos.Length; ++i)
{
    if (arquivos[i].Contains(nome) &&
        arquivos[i].Contains("shp"))
        return arquivos[i];
}
```

Com o nome do arquivo devolvido sabe-se agora qual a camada sobre a qual se vão trabalhar os dados e, assim sendo, adiciona-se esta ao mapa já existente no componente do MapWindow.

A informação existente no shapefile é armazenada localmente numa tabela na qual é adicionada uma coluna com os valores provenientes do DataSet enviado pelo cliente, para facilitar a manipulação da informação.

É possível agora preencher os polígonos com a cor estipulada para cada valor, conforme o código presente no Quadro 12:

Quadro 12 - Método de colorir de cada polígono

```
        UInt32 corBase;
        UInt32 corParaValorNulo =
Convert.ToUInt32(System.Drawing.ColorTranslator.ToOle(S
ystem.Drawing.Color.White));
        UInt32 corEfectiva;

        for (int i = 0; i < tabelaShapeFile.Rows.Count;
++i)
        {
            corBase =
Convert.ToUInt32(System.Drawing.ColorTranslator.ToOle(S
ystem.Drawing.Color.Navy));

            if
(System.Convert.ToInt32(tabelaShapeFile.Rows[i]["Valore
s"]) > 0)
                corEfectiva = corAInserir(corBase,
(uint)System.Convert.ToInt32(tabelaShapeFile.Rows[i]["V
alores"]), 5);
            else
                corEfectiva = corParaValorNulo;

            mapa.set_ShapeFillColor(0, i,
(uint)(corEfectiva));
        }
```

O método `corAInserir` atribui a cor através da criação de vários intervalos de valores, isto é, divide o valor máximo existente na tabela pelo número desejado de níveis que o usuário pretende. O valor de cada polígono cria uma variação da `corBase`, a qual é inserida no próprio polígono.

Caso o usuário pretenda visualizar os dados ao nível estadual do Brasil, conforme a Figura 17 representa, então o valor da string nome do Quadro 11 terá que ter o valor "Brasil".

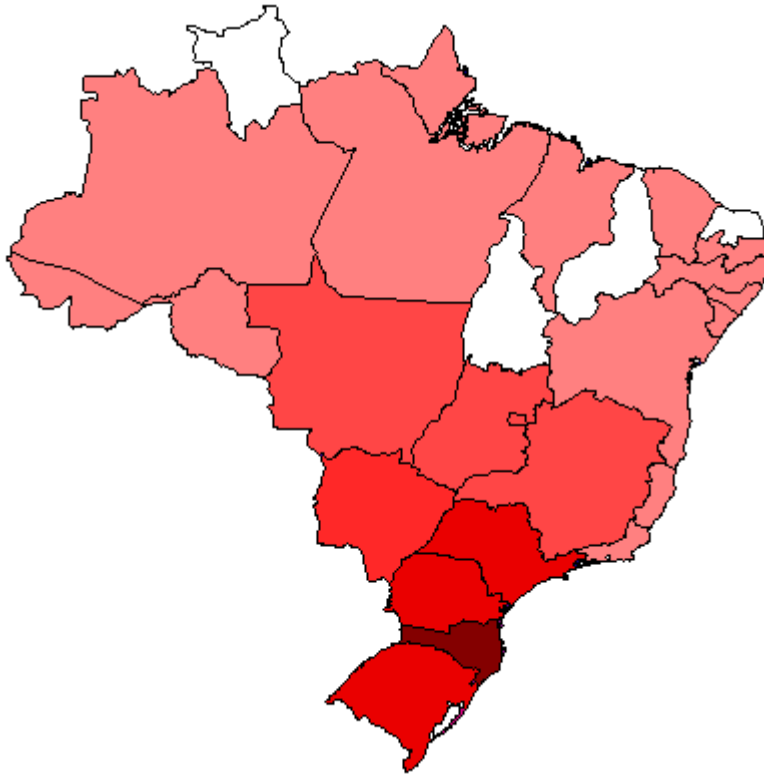


Figura 17 - Resultado de uma *query* por país

Outra opção que o usuário tem é efetuar uma consulta por Estado, isto é, pode escolher visualizar a informação por municípios dentro do estado selecionado. Neste caso o valor da primeira coluna da primeira linha da tabela enviada pelo cliente tem o nome do estado selecionado, por exemplo “Santa Catarina”, “Rio de Janeiro” etc..

A Figura 18 é a janela do protótipo, na qual o círculo representa o estado do servidor {ligado = Verde,desligado = Vermelho}, o botão permite ligar e desligar e à direita está o local em que o mapa é gerado e está pronto para ser enviado para o Web browser. A imagem do mapa refere-se a Santa Catarina.

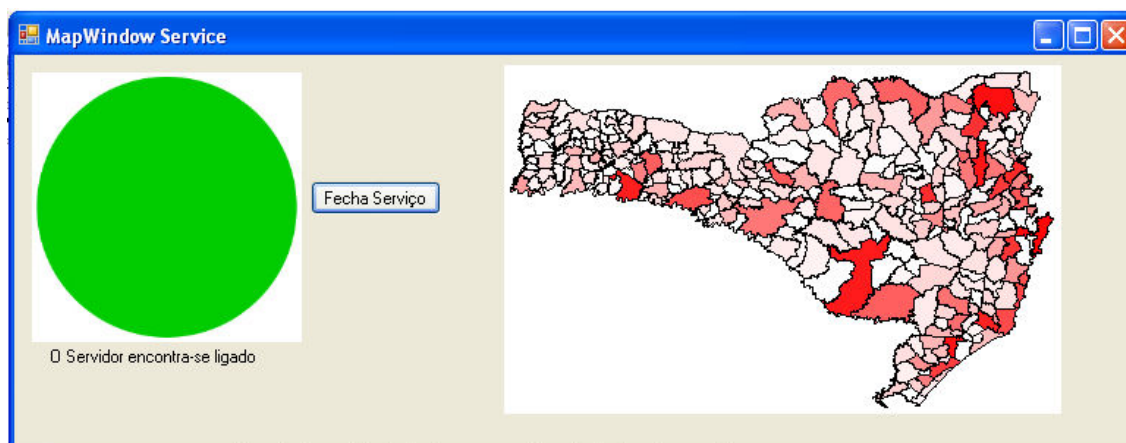


Figura 18 - Protótipo de aplicação de mapas

5.1.2 Servidor de mapas

Outra funcionalidade desta aplicação é o fato desta executar o papel de um servidor de mapas, visto que responde a um pedido com um mapa. Para proceder ao envio do mapa que foi gerado, este output é gravado no lado do servidor como uma imagem bitmap (.bmp) e, posteriormente, enviada para o cliente neste formato, para que possa ser mostrado no browser do lado do cliente. A imagem é guardada no formato bitmap devido a limitações do próprio MapWindow, que só permite que os mapas gerados sejam neste formato.

5.2 WebService

Para que exista uma ligação entre o servidor OLAP e o servidor de mapas foi implementado um Web Service para possibilitar essa comunicação.

De acordo com a Arquitetura proposta no capítulo 4, este componente da solução foi desenvolvido com quatro métodos:

1. recebeTabelaDados(in: Tabela com dados referentes às áreas geográficas)
 - o Este método é utilizado por parte do Web Browser, que fica

encarregado de enviar ao Web Service a tabela com os dados que quer mapear. O Web Service é responsável de receber a Tabela e guardá-la à espera de uma pergunta por parte do Servidor de mapas. Esta tabela possui duas colunas, onde a primeira é o nome da região (cidade ou estado) e a segunda possui os valores de cada uma.

2. `enviaTabelaDados(out: Tabela com dados referentes às áreas geográficas)`

- O Servidor de mapas consulta constantemente o Web Service a perguntar se existe alguma tabela de dados que possa mapear. Caso exista, recebe uma tabela de dados, senão recebe um valor *null*.

3. `recebeImagem(in: imagem)`

- Quando o servidor de mapas termina de mapear a tabela que recebeu anteriormente, pede ao Web Service para receber a imagem. Este, por sua vez, guarda-a consigo até que o browser faça o pedido pelo mapa.

4. `enviaImagem(out: imagem)`

- O Web Browser consulta constantemente o Web Service a perguntar se o mapa que requisitou previamente já está disponível. Caso o Web Service já tenha em sua posse o mapa, então envia a imagem correspondente para o browser.

A Figura 19 pretende dar uma melhor perspectiva de como funciona a relação do Web Service com o WebBrowser e o Servidor de Mapas. Este é o desenrolar lógico da comunicação, mas como exposto anteriormente, os métodos “`enviaTabelaDados`” e “`enviaImagem`” são invocados ciclicamente até enviarem a tabela com os dados ou a imagem, respectivamente.

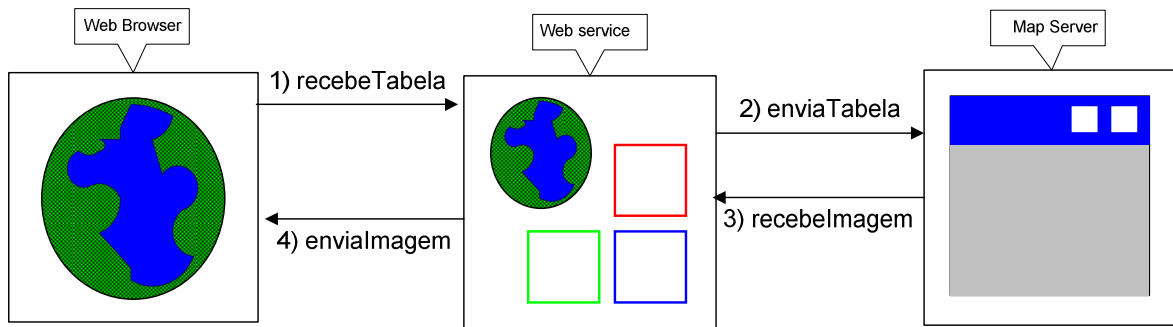


Figura 19 - Relação do Webservice com os outros componentes

6 Trabalhos relacionados

A maioria dos trabalhos relacionados com este estudo tenta definir padrões para a construção de sistemas OLAP+GIS completos, sugerindo *frameworks* para este propósito. Um grupo se destaca com um *framework* de código aberto baseado na web para processar dados geográficos e multidimensionais que é uma instância da arquitetura GOLAPA (*Geographical Online Analytical Processing Architecture*) [FIDALGO et al 2004]. Os autores propõem uma arquitetura completa para integração e visualização dos dados analíticos e geográficos, o que torna o projeto muito interessante. Porém, torna a aplicação restrita e de difícil evolução, isto é, fica difícil acompanhar as tecnologias mais recentes, pois isso implicaria em mudar a arquitetura do *framework* ou então mudar formatos e protocolos usados. Na solução proposta neste trabalho, por outro lado, cada componente, incluindo o servidor OLAP, *frontend*, e ferramenta GIS, pode ser substituído por um componente equivalente de outro fornecedor, desde que esse componente atenda aos padrões definidos.

Outra vertente de pesquisa na integração OLAP+GIS consiste na definição de uma linguagem diferente na manipulação dos dados diretamente no Data Warehouse, chamada SOLAP (Spatial OLAP) [RIVEST 2005]. Esta abordagem é bem consistente e já possui ferramentas muito completas que permitem ao usuário tornar mais eficaz o seu trabalho na visualização de informação geográfica a partir de um Data Warehouse. Apesar do estado avançado destas pesquisas, BIMONTE [2007] apresenta várias questões acerca da abordagem em si, como nas áreas da modelagem dimensional, visualização e medidas geográficas.

Maria Luisa Damiani [DAMIANI, 2004] defende um modelo dimensional espacial de dados para objectos de carácter geográfico. Esse modelo provém de um modelo multidimensional convencional com conceitos espaciais (facto espacial, dimensão espacial, medida espacial). Apesar de pertinente, a aplicação desta pesquisa só se torna

realmente útil na construção de Data Warehouses de raiz, pois no passado, como nos dias de hoje, poucos DWs são criados tendo em conta as dimensões geográficas.

Na Universidade Federal de Santa Catarina, o estudo da integração de ferramentas OLAP com GIS tem sido levado com muita seriedade, mas com uma abordagem diferente. O caminho das pesquisas tem sido acompanhado com os padrões de comunicação atuais e também com as tecnologias OLAP mais comuns, ou seja, as pesquisas na UFSC pretendem integrar as tecnologias OLAP e GIS com o mínimo de influência nos fundamentos destas. Os trabalhos desenvolvidos no Grupo de Banco de Dados da UFSC foram os seguintes:

- “Uma Aplicação OLAP sobre a Web para Análise dos Dados do Vestibular da UFSC e Diretrizes para a sua Integração com GIS” realizada por Felipe SHIGUNOV, no qual é dado mais ênfase ao aspecto OLAP, mas representa o início dos estudos voltados para a integração OLAP e GIS.
- “Desenvolvendo um Middleware para a Integração de Servidores OLAP e Servidores de Mapa na Web” realizado por Carlos Costa e José Neis, no qual “foi definida uma arquitetura aberta que combina componentes de mercado usando um middleware baseado em padrões de sistemas abertos das áreas de OLAP e SIG para realizar a ligação entre as duas tecnologias”.
- “Visualização cartográfica de informação acoplada a processamento analítico on-line”, realizado por Ricardo Gross, no qual explora com mais profundidade o estado da arte das ferramentas GIS.

7 Conclusões e trabalhos futuros

O trabalho aqui descrito apresentou um sistema desenvolvido para ambiente Web que permite a comunicação entre um Data Warehouse e o GIS através de XMLA, utilizando, principalmente, ferramentas da Microsoft. O protótipo criado permite a realização de consultas OLAP via Web e visualização dos resultados em mapas.

No início do projeto, considerou-se que seria mais fácil implementar o sistema com ferramentas proprietárias da Microsoft e que o conhecimento das mesmas seria de mais valia para o grupo de banco de dados da UFSC, visto que este vinha focando em ferramentas de código aberto. No entanto, foram encontradas algumas dificuldades de desenvolvimento, fazendo com que fosse necessário recorrer a novas soluções.

No caso do servidor de mapas, a solução foi um software de código aberto, o MapWindow GIS. Apesar desta ferramenta ser ainda um protótipo, tem grande flexibilidade e capacidades que fazem com que seja possível estudar a sua viabilidade comercial.

Este trabalho contribui para dar mais um passo na pesquisa acerca da integração OLAP+GIS, pois a ferramenta criada utiliza padrões nas pesquisas multidimensionais o que significa que se pode integrar facilmente com as tecnologias já existentes com poucos custos de integração.

Como trabalhos futuros, pode ser considerada a criação de uma interface que permita ao usuário manipular a imagem do mapa que está presente no web browser. Entre as funções que podem ser incluídas estão *zoom-in*, *zoom-out*, selecionar áreas do mapa para visualizar informação.

A definição de uma API formalizada e robusta, sob a forma de um Web Service, para a comunicação entre a parte OLAP e GIS, por sua vez, é um passo significativo para o

progresso no desenvolvimento desta temática. Parte desta melhoria é permitir concorrência (diversos usuários a recorrerem ao mesmo serviço ao mesmo tempo).

Referências Bibliográficas

ANOZELO, Cyntia Aurora Moura. **OLAP: conceito e sua Aplicação**, 2004. Diapositivos.

AVISMAP GIS: Engine. Disponível em:
<http://www.avismap.com/gis_products/avismap_gis_engine.html>. Acesso em:
15/11/2007.

BIMONTE, Sandro; TCHOUNIKINE, Anne; MIQUEL, Maryvonne. **Spatial OLAP: Open issues and a Web Based prototype**. Aalborg University: 10th Agile International Conference On Geographic Informations Science, 2007.

CARDOSO, Elsa: **Sistemas de Apoio à Decisão - Data Warehouse**, Instituto Superior das Ciências do Trabalho e da Empresa, 2006. Diapositivos.

Damiani M. L. **Spatial Data Warehouses**. Universidade de Milão. 2006

DT. Data WareHouse. Disponível em <http://pt.wikipedia.org/wiki/Data_warehouse>
Acesso em 27/08/2007.

DBQ. DBQ Consultant, Data WareHouse process:, Disponível em
<<http://www.gantthead.com/process/processMain.cfm?ID=9076>> Acesso em
10/09/2007

ESRI. ArcGIS Explorer. Disponível em:
<<http://www.esri.com/software/arcgis/explorer/index.html>>. Acesso em: 11/11/2007

HAROLD, Elliotte Rusty. **XML Bible2001**, John Wiley & Sons, Incorporated. 1600 páginas,
ISBN: 0764548190, 2001

FIDALGO R. N., TIMES V. C., SILVA J., SOUZA F.F., and SALGADO A.C. **Providing multidimensional and geographical integration based on a gdw and**

metamodels, In Brazilian Symposium on Databases (SBBD), 2004.

GIS. Geographic Information Systems, Disponível em
<http://pt.wikipedia.org/wiki/Geografic_Information_Systems> Acesso em 30/09/2007.

GROSS, Ricardo Wilmar. **Estudo de ferramentas de visualização cartográfica para acoplamento com OLAP sobre a Web**. IV Escola Regional de Banco de Dados, Florianópolis, 2008.

GOMES, José Cordeiro: **Sistemas de Informação para Gestão**, ISCTE 2005. Diapositivos.

IBGE. Instituto Brasileiro de Geografia e Estatística, Disponível em <http://www.ibge.gov.br/>, Acesso em 05/09/2007

IME. Introduction to Multidimensional Expressions., disponível em
<<http://tecnnet.microsoft.com/Library/MDX>>, Acesso em 17/09/2007

INMON, William: Como Construir o Data Warehouse, Wiley Editors, Rio de Janeiro, 1997

MACDONALD, Giovani; Rubik, João. **Pesquisa e Seleção de Ferramentas Livres e baseadas em padrões de sistemas abertos para a elaboração de interfaces OLAP sobre a Web**. Trabalho de Conclusão de Curso, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina 2007. 114 pags.

MAPPOINT: North America 2006. Disponível em:
<<http://www.microsoft.com/mappoint/products/2006/default.msp>>. Acesso em:
10/09/2007.

MAPWINDOW Disponível em: <<http://www.mapwindow.org/>>. Acesso em: 05/11/2007.

MD Modelo Dimensional: Disponível em <<http://>

imasters.uol.com.br/artigo/3836/modelo_dimensional_para_data_warehouse>,

Acesso a 27/04/2008

MELLO, João Alexandre Bonin de. **Uma proposta de modelo de dados para suporte ao processamento transactional e de Data Warehouse simultaneamente.**

Florianópolis, 2002.

MICROSOFT. SQL SERVER: Reporting services. Disponível em <<http://www.microsoft.com/sql/technologies/reporting/default.msp>>. Acesso em 01/06/2008.

MICROSOFT BRASIL. Disponível em <www.microsoft.com/brasil>, Acesso em 05/05/2008a

MICROSOFT BRASIL. SQL-Server. Disponível em <www.microsoft.com/brasil/sql> Acesso em 05/05/2008b.

NÉIS, José Filipe; da COSTA, Carlos Eduardo. **Desenvolvendo um Middleware para a Integração OLAP-SIG.** Trabalho de Conclusão de Curso, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, 2007.

NORTHRIDGE SYSTEMS, Disponível em <<http://www.northridge.com/BusinessIntelligenceTechnologyOverview.aspx>> Acessado em 05/06/2008

OLAP. Disponível em <<http://pt.wikipedia.org/wiki/OLAP>> Acesso a 20/09/2007

PENDSE, Nigel. OLAP Market Share Analysis, Disponível em <<http://www.olapreport.com/market.htm>> Acesso em 21/12/2007

QUANTUM GIS Disponível em: <www.qgis.org>. Acesso em: 20/11/2007

RIVEST, S., BÉDARD, Y., PROULUX, M.-J., NADEAU, M., HUBERT, F., PASTOR, J.,
2005 SOLAP: Merging Business Intelligence with Geospatial Technology for
Interactive Spatio-Temporal, 2005.

SHIGUNOV, Felipe: **Uma Aplicação OLAP sobre a Web para análise de dados do
Vestibular da UFSC e diretrizes para a sua integração com GIS**, Trabalho de
Conclusão de Curso, Departamento de Informática e Estatística, Universidade
Federal de Santa Catarina 2007, 88 pags.

SILVA, J.; TIMES, V.C.; Salgado, A.C. **An Open Source and Web Based framework for
Geographic and multidimensional Processing**. 2006.

TRIGUEIROS, Maria José: **Sistemas de Apoio à Decisão – OLAP**, ISCTE 2006.
Diapositivos.

WIG. What is GIS?. Disponível em <www.Gis.com>, Acesso em 30/09/2007.

XMLA, Disponível em <<http://www.xmla.org/>>, Acesso em 17/09/2007.