



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Silvestre Lomba Malta

“Business Intelligence” sobre informação espacial: um caso de estudo

Tecnologia e Gestão de Sistemas de Informação

Trabalho efetuado sob a orientação de
Doutor Jorge Manuel Ferreira Barbosa Ribeiro
Doutor Pedro Miguel Ribeiro de Castro

Outubro de 2012

Agradecimentos

Gostaria de agradecer muito particularmente à minha esposa Susana e meu filho Pedro pelo apoio que me deram ao longo desta jornada ao me ajudarem a concretizar um dos meus objetivos de vida.

Não poderia deixar de agradecer ao Professor Doutor Pedro Castro e Professor Doutor Jorge Ribeiro pelas boas orientações que me foram dando ao longo do desenvolvimento deste projeto, o meu obrigado a ambos.

Agradeço a todos os docentes e alunos envolvidos no estudo epidemiológico relativo ao perfil de saúde da população de Viana do Castelo, realizado pela Escola Superior do Instituto Politécnico de Viana do castelo em articulação com o Observatório de Saúde do Gabinete de Cidade Saudável, cujos dados foram utilizados como caso de estudo no âmbito deste projeto.

Aos meus colegas de mestrado, Rolando Costa e Marcelo Videira pela partilha de experiencias que tiveram comigo, foi muito bom o ambiente de camaradagem e de entre ajuda que tive com ambos.

Resumo

O trabalho desenvolvido neste projeto visou o estudo de ferramentas de business intelligence com a capacidade de analisar dados com uma componente geográfica. Após o estudo dos conceitos e ferramentas associadas à temática do business intelligence georeferenciado, optou-se por implementar uma solução utilizando soluções gratuitas e de código aberto. Como caso de estudo, foi desenvolvido um sistema de suporte à análise georreferenciada obtidos de um inquérito realizado aos residentes de um concelho de Portugal, permitindo proceder à descrição e análise do perfil de saúde da população.

Palavras-Chave: Business Intelligence, Sistemas de Informação Geográfica, Data Warehouse, Sistemas de Suporte à Decisão, Open Source

Abstract

The work developed in this project aimed the study of business intelligence tools with the ability to analyze data with a geographic component. After studying the concepts and tools related to the georeferenced business intelligence thematic, we choose to implement a solution using free open source solutions. As study case, it was developed a support system to the georeferenced analysis of the enquiries made by Escola Superior de Saúde of Instituto Politécnico de Viana do Castelo with the Healthy Cities Observatory Cabinet, to residents of a district of Portugal. The use of these data allowed the implementation of a support system to the georeferenced analysis of the enquiries allowing proceeding and analysis to the description of the health profile of the population.

Key-Words: Business Intelligence, Geographic Information Systems, Data Warehouse, Decision Support Systems, Open Source

ACRÓNIMOS

BI – Business Intelligence

SI – Sistemas de informação

TI – Tecnologias de informação

IPVC – Instituto Politécnico de Viana do Castelo

DW – Data warehouse

BD –Base dados

OLAP – On-Line Analytical Processing

SOLAP – Spatial On-Line Analytical Processing

ROLAP - Relational OLAP

ETL – Extract Transform Load

OLTP – On-Line Transaction Processing

ERP - Enterprise Resource Planning

SGBD – Sistema de Gestão Base Dados

SIG – Sistema de Informação Geográfica

GIS – Geographic Information System

OGC – Open Geospatial Consortium

WMS - Web Map Service

WFS - Web Feature Service

WCS - Web Coverage Service

WPS - Web Processing Service

GLS - Geographic Linkage Service

SLD - Styled Layer Descriptor

KML - Keyhole Markup Language

GML- Geography Markup Language

FSF - FreeSoftware Fondation

GPL - GNU Public Licence

OSI - Open Source Iniciative

CGI – Common Gateway Interface

API – Aplication Programming Interface

HTTP – HyperText Transfer Protocol

RDBMS - Relational Database Management System

DBMS - Database Management System

DEM – Digital Elevation Model).

CSS – Cascading Style Sheets

RIA – Rich Internet Aplication

KPI – Key Performance Indicators

Índice

1	Introdução	1
1.1	Enquadramento e Motivação	1
1.2	Objetivos	2
1.3	Escalonamento dos Trabalhos	2
2	Conceitos, Terminologias, Normas e Frameworks	5
2.1	Introdução	5
2.2	O que é o Business Intelligence e os Sistemas de Data Warehousing	5
2.3	O que é um Sistema de Informação Geográfico	7
2.4	Normas e Frameworks associadas aos SIG	9
2.4.1	Normas	9
2.4.2	Ferramentas comerciais e baseadas em software livre	12
2.5	Trabalho Relacionado	15
2.6	Reflexão Crítica	26
3	Caso de Estudo	27
3.1	Introdução	27
3.2	Contexto de aplicabilidade	27
3.3	Arquitetura Geral e Ferramentas Utilizadas	28
3.4	Características das Fontes de Informação	32
3.5	Construção do modelo multidimensional	38
3.6	O Processo de Extração, Transformação, e Carregamento	42
3.7	O Processo SOLAP	58
3.8	Business Intelligence – Visualização de Dados Analíticos	62
3.9	Análise e Discussão	70
4	Conclusão e Trabalho Futuro	73
5	Referências Bibliográficas	75
6	Anexos	77
6.1.1	Alterações introduzidas ao testpage.jsp	77
6.1.2	Ferramentas Comerciais	82
6.1.3	Ferramentas Baseadas em Tecnologia Livre	88

Índice de figuras

Figura 1 – Ilustração da representação da Informação Geográfica em Camadas/Layers. Fonte: (Pinto, s.d.)	8
Figura 2 - Relacionamento entre normas OGC (adaptado de Rita, 2010).	12
Figura 3 - Esquema de visualização de dados. Fonte: (Wen Wang, 2010).....	16
Figura 4 - Ponto de Vista Conceptual. Fonte: (Klaus Bohm, 2011).....	18
Figura 5 - Infraestrutura com Geo extensões. Fonte (Klaus Bohm, 2011)	18
Figura 6 - Arquitetura de Widgets Fonte (Klaus Bohm, 2011).....	19
Figura 7 - Comparação de Ferramentas Open Source. Fonte (Bernardino, 2011)...	20
Figura 8- Arquitetura do Sistema. Fonte (Wang Lianbo, 2010).....	21
Figura 9 - Aplicações Open Source Escolhidas. Fonte (Wang Lianbo, 2010)	23
Figura 10- Arquitetura do Sistema. Fonte (LI Guangyu, 2010)	24
Figura 11 - Exemplo de Código para publicação de Mapas usando OpenLayers. Fonte (LI Guangyu, 2010)	25
Figura 12 - Demonstração de Código de Event Click num Mapa. Fonte (LI Guangyu, 2010).....	26
Figura 13 – ETL, OLAP e Ferramentas de BI.	29
Figura 14 - Amostra de dados fornecidos no Excel.	35
Figura 15 - Shapefile pontos.	36
Figura 16 - Shapefile Freguesias.....	37
Figura 17 - Shapefile Secções.	37
Figura 18 - Shapefile Sub Secções.	38
Figura 19 - Modelo Multidimensional	40
Figura 20 - Job inicial.	43
Figura 21 - Leitura distinta de a0803	44
Figura 22 - Alimentação de tabelas de dimensão	45
Figura 23 - Transformação Excel-to-DW.	45
Figura 24 - Excel Input.	46
Figura 25 - Mapeamento da doença Diabetes.....	47

Figura 26- Mapeamento do Estado Civil.....	48
Figura 27 - Operações de casting.	49
Figura 28 - Table Output.	50
Figura 29 - Sql Editor.	50
Figura 30 - Mapeamento de valores.....	51
Figura 31 - Mapeamento manual.....	52
Figura 32 - Transformação Pontos-to-DW.....	53
Figura 33 - Leitura do shapefile.	53
Figura 34 – Seleção do Sistema de Coordenadas.....	54
Figura 35 - Renomeação de campos.....	54
Figura 36 - Update na tabela de factos.....	55
Figura 37 - Transformação Freg-Seccao-to-DW.....	55
Figura 38 - Table output na tabela de dimensão d_seccoes.....	56
Figura 39 Atualização de tabela de factos.....	57
Figura 40 - Tomcat fazendo o deploy do geomondrian.war.	58
Figura 41 - Schema Workbench.	60
Figura 42 - Excerto XML do Cubo	62
Figura 43 - Raiz do website GeoMondrian.	62
Figura 44 - Consulta MDX.....	63
Figura 45 - Consulta efetuada no JPivot.....	64
Figura 46 - Open OLAP Navigator.....	65
Figura 47 - Consulta MDX devolvida pelo OLAP Navigator	65
Figura 48 - Visualização de multi-poligono da dimensão Geo.	66
Figura 49 - Extração de Polígonos e Quantidades	67
Figura 50 - Representação Geográfica da Informação	68
Figura 51 – Efeito Drill-Down.....	68
Figura 52 - Efeito Roll-Up e Drill-Down.....	69
Figura 53 – Efeito Drill-Down em todas as Freguesias	70

1 Introdução

1.1 Enquadramento e Motivação

Nestes últimos anos verifica-se que ganham, pelo interesse, número e resultados potenciais e reais, os projetos de sistemas de informação geográfica (SIG). Os SIG revelam-se como ferramentas com um grande potencial no suporte aos processos de tomada decisão por permitirem adotar metodologias inovadoras na análise de informação, sendo, atualmente, explorados por um número crescente de utilizadores numa diversidade de áreas, muito por força das suas características de representação e análise de dados georreferenciados.

Neste contexto, a motivação deste trabalho centra-se no estudo das vantagens associadas aos tradicionais Sistemas de Business Intelligence e a sua aplicabilidade a dados espaciais no sentido de estudar a viabilidade da criação de uma ferramenta de análise analítica e GeoEspacial que potencie a utilização e exploração dos dados provenientes de um sistema SIG.

Este projeto centra-se no estudo, análise e exploração dos conceitos associados aos Sistemas de Suporte à Decisão, em particular as ferramentas “Bussiness Intelligence”, que permitem explorar a componente espacial dos dados utilizados nos Sistemas de Informação Geográfica. O estudo irá ser aplicado a dados resultantes do trabalho de recolha para o diagnóstico do perfil de saúde de um concelho de Portugal no âmbito do protocolo entre a Escola Superior de Saúde do Instituto Politécnico de Viana do Castelo e o Observatório de Saúde do Gabinete de Cidade Saudável.

1.2 Objetivos

Este trabalho pretende estudar e aplicar conceitos, metodologias, e ferramentas no sentido de testar e consolidar as temáticas abordadas nas várias unidades curriculares do curso. Neste sentido este projeto visa o estudo de ferramentas de Business Intelligence com a capacidade de analisar dados com uma componente geográfica, incluindo uma revisão à literatura nesta área e a utilização, aplicação e teste de uma ou mais ferramentas a selecionar sobre os dados de um inquérito realizado pela Escola Superior de Saúde do Instituto Politécnico de Viana do Castelo e o Observatório de Saúde do Gabinete de Cidade Saudável. De salientar que no desenvolvimento do projeto serão preferencialmente utilizadas soluções gratuitas e de código aberto.

1.3 Escalonamento dos Trabalhos

O método experimental que se utilizou para a realização desta dissertação teve como sustentação os passos metodológicos que se iniciaram, pela identificação do problema, no sentido de formalizar um objetivo, sobre o qual, o mesmo foi desenvolvido. Subsequentemente, a informação foi recompilada, organizada e analisada continuamente, construindo uma proposta para resolver o problema identificado (Secção 3). Finalmente, foram elaboradas conclusões que refletiram os resultados obtidos na avaliação da operacionalização. Neste sentido este trabalho foi desenvolvido em quatro fases:

1. Revisão da literatura existente sobre os paradigmas utilizados - i.e. tornou-se essencial apresentar os conceitos, metodologias e referencias que sustentaram este estudo, a revisão e resumo da bibliografia, assim como dos mais recentes trabalhos que abordam a área do saber deste trabalho, nomeadamente: Business Intelligence, a construção de modelos multidimensionais, processos de extração, transformação e carregamento de dados em data warehouse e a georreferenciação de dados contidos em Spatial Online Analytical Processing (SOLAP).

2. Levantamento dos requisitos (ou Análise dos procedimentos) – após uma análise profunda dos conhecimentos adquiridos na fase anterior efetuou-se um levantamento de toda a estrutura e procedimentos das Sistemas de Informação (SI) / Tecnologias de Informação (TI) de forma a verificar as soluções disponíveis e em que estado é que se encontram.
3. Aplicabilidade de Georeferenciação em OLAP – esta fase de índole mais prática permitirá materializar as ideias conceptualizadas na primeira fase.
4. Avaliação – inclui a análise dos resultados alcançados a fim de verificar que o objetivo de georeferenciar dados obtidos em OLAP foi alcançado.

A tabela 1 identifica o cronograma das fases de execução deste trabalho.

FASE	Mês								
	F	M	A	M	J	J	A	S	O
1. Revisão da literatura existente sobre os paradigmas utilizados	X	X	X						
2. Levantamento dos requisitos / Análise dos procedimentos			X						
3. Utilização, aplicação e teste de uma ou mais ferramentas a selecionar num inquérito realizado pela Escola Superior de Saúde do Instituto Politécnico de Viana do Castelo e o Observatório de Saúde do Gabinete de Cidade Saudável				X	X	X	X		
4. Avaliação							X	X	X
5. Escrita do documento do projeto de Mestrado.							X	X	X

Tabela 1- Organização cronológica das fases da dissertação.

O trabalho foi desenvolvido em cerca de dez meses, nos primeiros quatro meses foram centrados na revisão de literatura existente, assim como no levantamento dos requisitos. Nos cinco meses finais foram realizadas as tarefas de aplicação das ferramentas escolhidas e da sua avaliação. Por fim, os últimos três meses foram destinados para a escrita deste documento.

2 Conceitos, Terminologias, Normas e Frameworks

2.1 Introdução

Atualmente as empresas de médio e grande porte possuem departamentos de Tecnologias de Informação próprios ou trabalham com empresas externas que as auxiliam na gestão das Tecnologias de Informação (TI). A verdade é que estes dados se forem devidamente tratados por sistemas de Business Intelligence (BI) podem auxiliar os gestores das empresas a fundamentar de forma mais consolidada e com mais conhecimento as tendências de mercado bem como o conhecimento no seu ramo de negócio. Para tal os departamentos de TI devem recorrer a ferramentas apoio à tomada de decisão. Existem no mercado enumeras ferramentas que podem extrair conhecimento nas bases de dados dos softwares de gestão na sua globalidade. No âmbito desta tese, foi efetuado um levantamento de ferramentas de BI e de Sistemas de Informação Geográfica GIS.

A implementação de um sistema de apoio à decisão, requer um esforço com vista à obtenção de uma solução a preços de mercado competitivo. Uma das soluções passa pela escolha de ferramentas Open Source o que diminui despesas e grandes investimentos.

Este capítulo apresenta um levantamento de normas, ferramentas comerciais e não comerciais mais abrangentes e conhecidas no mercado. A pesquisa focou-se essencialmente em ferramentas de Business Intelligence aplicadas à informação espacial.

2.2 O que é o Business Intelligence e os Sistemas de Data Warehousing

Business intelligence tem como objetivo apoiar e melhorar a tomada de decisões. Refere-se principalmente em técnicas baseadas em computador. Essas técnicas são utilizadas para identificar, extrair e analisar dados de negócios. As tecnologias de BI fornecem históricos, pontos de vista atuais e perspectivas das operações de negócios. As funções de tecnologia de BI

fornece aos tomadores de decisão dashboards, relatórios, scoreboarding, mineração de processos, mineração de dados, análise de gestão de desempenho, processamento analítico online (OLAP), análise de negócios e análise preditiva. OLAP é uma categoria de ferramentas de apoio à decisão, muitas vezes utilizados para fornecer acesso, de forma eficiente e intuitiva, a um data warehouse ou "datamart". Na área de data warehousing, geralmente ouvimos falar de discussões sobre que filosofia abordar, normalmente esta discussão divide-se entre duas teorias, a de Bill Inmon, e a de Ralph Kimball (1KeyData, s.d.). Segundo o paradigma de Bill Inmon, um data warehouse corresponde a uma parte do sistema global de business intelligence, em que uma empresa constrói data marts com informações obtidas a partir desse repositório de dados. Segundo o paradigma de Ralph Kimball, o data warehouse corresponde a um aglomerado de data marts caracterizados segundo as áreas ou departamentos de uma empresa ou dos vários agentes de decisão de uma organização. No entanto, apesar de serem duas visões ou abordagens abrangentes e consensuais do conceito de data warehouse, ambas seguem o modelo dimensional. Na realidade, os data warehouse na maioria das empresas estão mais próximos do paradigma de Ralph Kimball. Isto porque a maioria dos data warehouses começaram como um esforço do departamento de TI, e portanto originaram-se como se fossem um data mart. Somente quando data marts adicionais foram construídos é que a arquitetura de dados evoluiu para um verdadeiro data warehouse. Ambos concordam que é necessário existir uma arquitetura de dados, as dimensões e medidas devem ser partilhadas por todas as áreas da empresa, o esquema em estrela é útil para apresentar informações aos utilizadores, o data warehouse deve ser construído iterativamente, os metadados são fundamentais e a implementação do processo de ETL deve ser automático. (OGC, 2011).

Em suma, um Data Warehouse é caracterizado pelas seguintes características [Inmon, 1996]:

- A informação está estruturada e integrada por assuntos;
- A informação está etiquetada temporalmente;
- A informação é não volátil;

Tipicamente, a necessidade do desenvolvimento de um Data Warehouse surge quando o problema da utilização dos Sistemas de Informação se centra no desempenho aquando das análises de informação nos sistemas de suporte à decisão, em particular quando os sistemas transacionais não suportam ou degradam o desempenho na execução de consultas complexas, nomeadamente quando envolvem várias variáveis de análise.

Por sua vez, o objetivo principal das ferramentas de BI é auxiliar os utilizadores a adquirir uma melhor compreensão do contexto comercial e do mercado onde se enquadra, disponibilizando meios rápidos e expeditos em termos de análises analíticas contemplando as características da informação armazenada num data warehouse.

O objetivo principal do BI é ajudar a empresa a adquirir uma melhor compreensão do contexto comercial e do mercado onde se enquadra. Para auxiliar a empresa nestes objetivos, o BI realiza uma coleta, integração, análise, interpretação e apresentação de informações de negócio.

2.3 O que é um Sistema de Informação Geográfico

Consoante o contexto de utilização (Pinto, s.d.) podemos definir um SIG como sendo um “Conjunto de procedimentos, manual ou automatizado, utilizados no sentido do armazenamento, e manipulação de informação georreferenciada.” Aronoff (1989) descreve um SIG como um “Sistema de apoio à decisão envolvendo integração de informação georreferenciada num ambiente de resolução de problemas.” De forma mais redutora Cowen (1988), define que um SIG é um conjunto de “Ferramentas com avançadas capacidades de modelação geográfica.” Outros autores caracterizam de forma mais abrangentes um SIG é um “Conjunto de funções automatizadas, que fornecem aos profissionais, capacidades avançadas de armazenamento, acesso, manipulação e visualização de informação georreferenciada.” (Koshkarirov, s.d.), (Azemoy, Smith e Sicherman, 1981)

Num SIG a informação geográfica é organizada em camadas ou níveis de informação (layers), consistindo cada uma num conjunto selecionado de

objetos associados e respetivos atributos (Figura 1). Existem vários modelos de dados aplicáveis em SIG. O SIG separa a informação em diferentes camadas temáticas e armazena-as independentemente, permitindo trabalhar com elas de modo rápido e simples, permitindo ao operador ou utilizador a possibilidade de relacionar a informação existente através da posição e topologia dos objetos, com o fim de gerar nova informação. OpenLayers é o exemplo de uma aplicação gratuita que facilita a colocação de um mapa dinâmico em qualquer página web. É possível com o OpenLayers, exibir excertos de mapas e marcadores carregados a partir de qualquer fonte. OpenLayers foi desenvolvido para promover o uso de informação geográfica de todos os tipos. OpenLayers é uma biblioteca Javascript livre e Open Source e está sob a cláusula 2-da licença BSD (também conhecido como o FreeBSD).

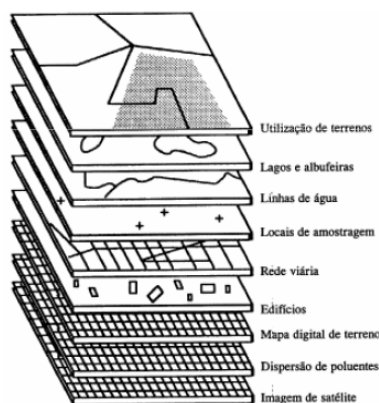


Figura 1 – Ilustração da representação da Informação Geográfica em Camadas/Layers. Fonte: (Pinto, s.d.)

Os modelos mais comuns em SIG são o modelo raster ou matricial e o modelo vetorial. O modelo de SIG matricial centra-se nas propriedades do espaço, compartimentando-o em células regulares (habitualmente quadradas, mas podendo ser retangulares, triangulares ou hexagonais). Cada célula representa um único valor. Quanto maior for a dimensão de cada célula (resolução), menor é a precisão ou detalhe na representação do espaço geográfico.

No caso do modelo de SIG vetorial, o foco das representações centra-se na precisão da localização dos elementos no espaço em que os objetos são reduzidos a um conjunto de coordenadas (x,y) ou (x,y,z). Para modelar

digitalmente as entidades do mundo real utilizam-se essencialmente três formas espaciais: o ponto, a linha e o polígono.

O SIG pode funcionar como uma base de dados com informação geográfica (dados alfanuméricos) que se encontra associada por um identificador comum aos objetos gráficos de um mapa digital. Desta forma, assinalando um objeto pode-se saber o valor dos seus atributos, e inversamente, selecionando um registro da base de dados é possível saber a sua localização e apontá-la num mapa (Wikipedia, s.d.).

2.4 Normas e Frameworks associadas aos SIG

2.4.1 Normas

Normas são documentos técnicos que contêm os detalhes ou codificações técnicas. Os programadores de software usam estes documentos para a construção de interfaces e codificações abertas nos seus produtos e serviços. Esses padrões são os principais "produtos" do OGC (Open Geospatial Consortium) e foram desenvolvidos pela associação para enfrentar os desafios de interoperabilidades específicas. Idealmente, quando os padrões OGC são implementados em produtos ou serviços on-line por dois engenheiros de software diferentes que trabalham de forma independente, os componentes finais programados, conseguem comunicar de uma forma plug and play sem que seja necessária depuração adicional. Esta técnica está baseada na chamada especificação abstrata - descrevendo um modelo de dados básico para características (features) geográficas a serem representadas.

A organização OGC é uma organização voluntária internacional de padrões de consenso. Na OGC, 479 organizações comerciais, governamentais, não-lucrativas e instituições de pesquisa do todo o mundo, colaboram num processo de consenso aberto encorajando o desenvolvimento e a implementação de padrões e normas para conteúdo e serviços SIG.

Algumas das especificações mais relevantes no contexto do desenvolvimento de aplicações web são:

WMS - Web Map Service: corresponde à especificação de um formato, que tem como objetivo colocar uma mapa no formato de imagem podendo ser armazenado num ficheiro ou apresentado num ficheiro ou página na Internet.

WFS - Web Feature Service: é um dos serviços especificados pela OGC para acesso e manipulação de dados geográficos na Web. Este serviço permite o acesso a dados alfanuméricos e geográficos de temas vetoriais independentemente do formato de armazenamento.

WCS - Web Coverage Service: é um serviço que fornece comunicação eletrónica baseada na arquitetura cliente/servidor de dados geográficos. Estas informações são existentes sob a forma de coberturas multidimensionais. Estas são compostas por valores ou propriedades referentes as localizações geográficas espaçadas de forma regular através de um, dois ou três eixos de um sistema de coordenadas geográfico, podendo também conter informação temporal, regular ou irregularmente espaçada. O WCS também pode fazer o tratamento de dados modelados como geo-campos, em complementação ao serviço WFS, que trata de dados modelados como geo-objetos, isto é, que representam entidades espaciais discretas e bem definidas.

WPS - Web Processing Service: é uma interface standard que disponibiliza as regras de como são realizados os inputs e outputs na invocação de serviços de processamento geoespacial. O standart WPS define como um cliente pode requerer a execução de um processo e como são tratados os outputs desse processo. Define uma interface que facilita a publicação de processos geoespaciais e a descoberta aos clientes de como se ligarem a esses processos. WPS permite o acesso a web services usando HTTP GET, HTTP POST bem como através de SOAP, permitindo ao cliente o uso do mecanismo mais apropriado.

GLS - Geographic Linkage Service: destina-se a fornecer uma forma simples de troca de dados que contenha informação geográfica mas que não inclua os detalhes geométricos do objeto. Estes tipos de dados são denominados de Framework Dataset.

SLD - Styled Layer Descriptor: é uma especificação do OpenGIS Web Map Service (WMS) que define uma codificação que estende o standard WMS. Esta extensão permite uma codificação de símbolos e cores definidas pelo programador passíveis de serem usados para desenhar dados nos mapas geográficos. É necessário utilizar uma linguagem de estilos que tanto o cliente como o servidor consigam entender. SLD é tipicamente usado para instruir o WMS de como realizar a renderização de determinada layer.

KML - Keyhole Markup Language: é uma linguagem baseada em XML e serve para expressar anotações geográficas e visualização de conteúdos existentes nessa linguagem sobre a forma de mapas em 2D e navegadores terrestres em 3D. KML foi desenvolvido para uso com o Google Earth, que era originalmente chamado de Keyhole Earth Viewer. Este foi criado pela Keyhole, Inc, e que mais tarde foi adquirida pelo Google em 2004.

GML - Geography Markup Language: é uma gramática XML para descrever entidades geográficas. Pode servir de linguagem de modelação para sistemas geográficos para a troca de informação geográfica. O seu propósito é a codificação de dados geográficos, incluindo propriedades geométricas e não geométricas, a fim de permitir a troca, integração, partilha e visualização de informação geográfica

Os Schemas (xsd, xslt, etc) que são suportados, mantidos e cuja implementação esteja de acordo com as normas da OGC, podem ser consultados e feito o seu download no sitio oficial <http://schemas.opengis.net/>.

A figura 2 (adaptada de Rita, 2010), representa graficamente as relações entre diversas normas.

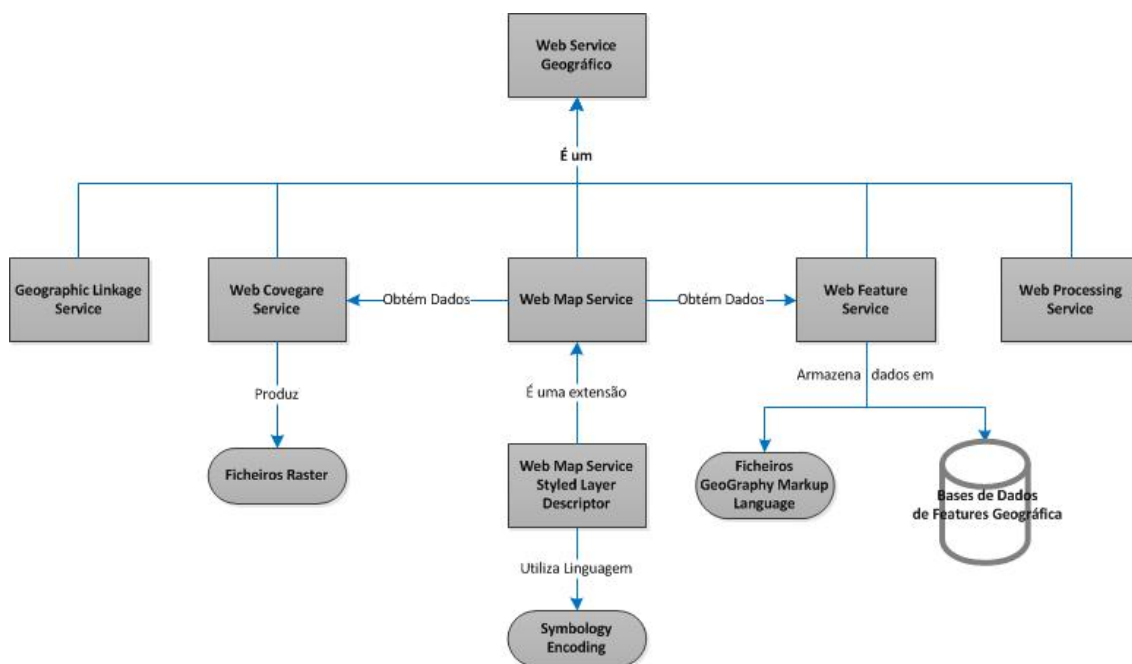


Figura 2 - Relacionamento entre normas OGC (adaptado de Rita, 2010).

2.4.2 Ferramentas comerciais e baseadas em software livre

Tradicionalmente o Open Source é associado com a dificuldade da manutenção. No entanto nos últimos anos tem crescido o número de utilizadores e tem mudado a sua “conotação”. O software livre e de código aberto, são cada vez mais uma alternativa ao uso de software comercial. As principais razões tem a ver com a diminuição de encargos com licenciamento, uma melhoria substancial da qualidade do software bem como da sua manutenção. Tal como as restantes comunidades de open source, os open SIG são compostas por centenas de pessoas e empresas que tem como objetivo comum apoiar os Open SIG (Ficher, s.d.). Existem duas organizações principais, a Open Source Geospatial Foundation (OSGeo) e Open Geospatial Consortium (OGC). A OSGeo é uma organização sem fins lucrativos que suporta e promove o desenvolvimento de dados e tecnologia open geospatial. OGC é um consórcio industrial internacional composto por 480 empresas, agências governamentais e universidades que trabalham em conjunto para desenvolver e disponibilizar especificações de interfaces públicas.

Existe uma enorme variedade de softwares que estão disponíveis gratuitamente, no entanto é preciso distinguir quais são as diferenças entre software comercial, livre, gratuito e de código aberto.

Os softwares comerciais são softwares que possuem uma licença comercial, que na maioria das vezes é paga. Esta licença garante uma licença de direito de uso do software, portanto não garante a compra do software propriamente dito.

Software livre nasceu de um movimento social que defende uma causa, causa essa que começou a tomar forma em 1983 pelas mãos de Richard Stallman, sendo criado na época o GNU que é um sistema operativo do tipo Unix. Em 1991 quando este sistema operativo estava praticamente pronto, faltando apenas o núcleo (kernel) do sistema operativo, Linus Torsvald criou um núcleo que poderia usar todos os módulos do sistema operativo GNU (Gnu, s.d.), ficando este núcleo conhecido como Linux. Deste movimento social, nasceu a FreeSoftware Foundation (FSF) que é uma entidade sem fins lucrativos que serve de base para apoiar e servir o movimento de software livre. Software livre é uma questão de liberdade, não de preço. Para entender este conceito, deve-se pensar em liberdade no sentido de “liberdade de expressão”, e não de “gratuito”. O utilizador tem liberdade não só para utilizar, bem como copiar, distribuir, modificar e estudar o software . Foram criados quatro fundamentos que servem de base: (Wikipédia, 2013)

- A liberdade de executar o programa, para qualquer propósito (liberdade nº 0).
- A liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades (liberdade nº 1). Acesso ao código-fonte é um pré-requisito para esta liberdade.
- A liberdade de redistribuir cópias de modo a ajudar terceiros (liberdade nº 2).
- A liberdade de aperfeiçoar o programa, e disponibilizar os seus aperfeiçoamentos, de modo que toda a comunidade beneficie (liberdade nº 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

A GNU Public Licence (GPL) é uma licença criada pela FSF que se baseia nos quatro fundamentos que a entidade defende. Quando um autor ou entidade disponibilizam um software que possui licença GPL, significa que é realmente um software livre. Para evitar que software livre seja modificado e disponibilizado mediante uma série de restrições foi criado o conceito de copyleft que garante que as garantias de modificação e distribuição são garantidas nos projetos derivados de software livre.

Software gratuito (freeware) é um software que pode ser utilizado gratuitamente mas cujo código fonte é fechado. Um software nestas condições é restrito na medida em que apenas o autor ou entidade que o desenvolve é que tem acesso ao seu código fonte.

O Open Source é um software que respeita as quatro liberdades definidas pela FSF. A nomenclatura e discurso Open Source foram criadas por Eric Raymond e outros fundadores da Open Source Initiative (OSI) com o objetivo de apresentar software livre às empresas de uma forma mais comercial, evitando desta forma o discurso ético do software livre da FSF que se baseia nas questões éticas de direitos e liberdades enquanto a OSI usa o termo Open Source sob um ponto de vista puramente ético. A OSI não ignora as liberdades da FSF mas torna-as mais flexíveis. Para que um software seja considerado Open Source deve respeitar dez requisitos (OpenSource, s.d.):

1. Distribuição livre;
2. Acesso ao código-fonte;
3. Permissão para criação de trabalhos derivados;
4. Integridade do autor do código-fonte;
5. Não discriminação contra pessoas ou grupos;
6. Não discriminação contra áreas de atuação;
7. Distribuição da licença;
8. Licença não específica a um produto;
9. Licença não restritiva a outros programas;
10. Licença neutra em relação à tecnologia.

2.4.2.1 Vantagens e desvantagens do software livre

Economicamente o software livre promove parcerias entre diversos parceiros associando esforços e diminuindo custos, mas também promove a competitividade entre os diversos agentes económicos. Esta competição traz vantagens aos utilizadores na melhoria de serviços de suporte. (GHOSH, 2006).

Uma das principais vantagens do software livre é a partilha do código fonte, esta partilha simplifica o desenvolvimento de novas aplicações. Devido ao código poder ser reaproveitado existe uma diminuição de custos e duplicação de esforços no desenvolvimento. Além disto o facto de se alterar código reflete-se numa melhoria do mesmo, pois o número de programadores envolvidos é maior, logo a probabilidade de identificação e correção de código aumenta, aliando ainda o facto deste esforço ser realizado num tempo menor. A qualidade do código tem tendência para aumentar pois os programadores são mais cuidadosos e criteriosos na elaboração do código, devido ao facto de saberem que este será avaliado por outros profissionais. (SABINO & KON, 2009)

2.5 Trabalho Relacionado

Nas últimas décadas tem crescido o número de investigadores a debruçarem-se sobre a utilização de tecnologias SIG em várias áreas de aplicabilidade. Fruto deste crescente aumento de estudos a literatura sobre a aplicabilidade dos SIG tem aumentado. *Wen Wang e Huijuin Luo* 2010, publicaram um artigo onde abordam a temática WebGIS. Desenharam uma plataforma para monitorizar dados públicos relacionados com a qualidade do ambiente. A plataforma desenhada inclui uma base de dados, um servidor WebGIS e um cliente Web. O artigo aborda alguns servidores WebGIS Open Source. Foi utilizada a aplicação MapServer CGI e Map-Script para aceder aos dados espaciais armazenados numa base de dados PostgreSQL com extensões postGIS. Os dados alfanuméricos a monitorizar estavam armazenados numa base de dados, enquanto que o armazenamento dos

dados espaciais distribuía-se por uma base de dados e ficheiro em formato shapefile. Relativamente aos mapas e visualização de dados, a plataforma continha dois componentes, o WebGIS Server para dar suporte à visualização dos mapas e o web cliente para permitir um interface para o utilizador. O servidor WebGIS é utilizado para aceder aos dados espaciais de forma a fornecer o serviço de Web Map Service (WMS) para a visualização dos dados geo-espaciais. Com a aplicação MapServer CGI e utilização do MapScript o Apache fornece capacidades de navegação aos serviços de GIS com WMS ou imagens. A navegação dos dados geospaciais por parte do utilizador é fornecida pelas bibliotecas WebGIS permitindo funcionalidades como focar, mostrar ou esconder layers, pesquisar dados ambientais e localizar pontos de monitorização nos mapas.

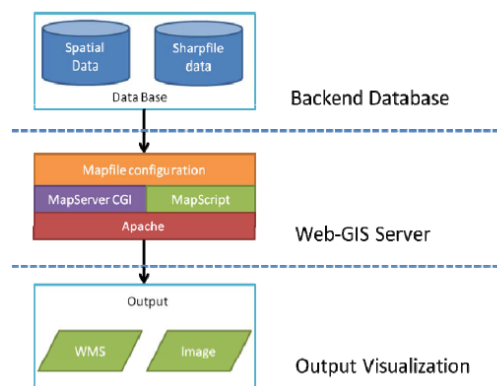


Figura 3 - Esquema de visualização de dados.

Fonte: (Wen Wang, 2010)

A arquitetura da plataforma consiste em três componentes: a monitorização ambiental e seus dados espaciais, o servidor WebGIS e o cliente Web. A base de dados é suportada pelo PostgreSQL com PostGIS. O PostGIS é uma ferramenta poderosa no manuseamento de dados espaciais bem como na conversão de ficheiros shapefile. O servidor WebGIS fornece o serviço de WMS para os clientes Web, tendo sido utilizado o Apache como WebServer. O MapServer é um motor Open Source de renderização, permite pesquisas raster, vetoriais, e baseadas em bases de dados, permite também linguagens de programação tais como PHP, Python, Perl, Ruby, Java e .Net. O Mapserver

accede aos dados espaciais usando um ficheiro de configuração (mapfile) onde estão definidas as relações entre os objetos, apontando ao MapServer onde estão localizados os dados e definindo como serão desenhados. Através do Mapfile, a aplicação MapServer CGI exhibe o mapa ao cliente Web. O Web Client foi construído com OpenLayers e PHP. O OpenLayers é uma biblioteca escrita em JavaScript que permite a visualização de mapas. Fornece uma API JavaScript que facilita a incorporação de mapas de variadíssimas origens dentro de uma página web ou aplicação. O cliente Web, disponibiliza o web-map fornecido pelo servidor WebGIS bem como a sua gestão (navegação, botões, icons, marcadores e seleção de layers) sendo a programação realizada com uma combinação de PHP e Javascript.

Klaus Bohm, Anett Mehler-Bicher e Dennis Fenchel, publicaram um artigo intitulado GeoVisual Analytics in the Public Health Sector (Klaus Bohm, 2011) onde abordam a temática de BI, análise GeoVisual e cloud computing com soluções Open Source. A aproximação escolhida, foi uma solução de BI com serviços GIS que pudesse ser hospedada na cloud. Do ponto de vista conceptual a camada mais baixa do sistema pode ser alojada na cloud e a camada intermédia representa os modelos funcionais com software Open Source de BI com componentes GIS. Foram estudados vários softwares de BI, mais propriamente, JaperSoft, Palo e Pentaho. Foi dada especial atenção à performance, flexibilidade e potencial de integração de dados geo-espaciais. O estudo revelou que a melhor ferramenta seria o Pentaho. Esta ferramenta já havia sido escolhida por *Dube e Badar* (Tomaszewski, 2009) em que apresentaram uma solução de arquitetura BI com componentes geo espaciais.

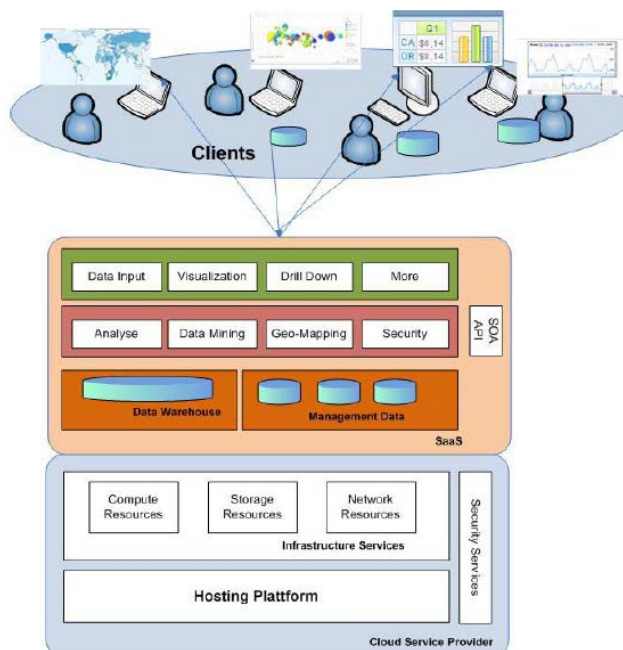


Figura 4 - Ponto de Vista Conceptual. Fonte: (Klaus Bohm, 2011)

Dube e Badar (Klaus Böhm, 2011) usaram o GeoKettle para a importação de dados e o GeoMondrian para aceder aos dados BI com referencias geoespaciais. A data warehouse foi extendida com funcionalidades de bases de dados espaciais, com estas funcionalidades implementadas fazendo uso de PostgreSQL com extensões PostGIS. De notar que deram especial atenção à aplicação GeoMondrian de forma a garantir que poderia conectar-se à sua Rich Internet Application (RIA).

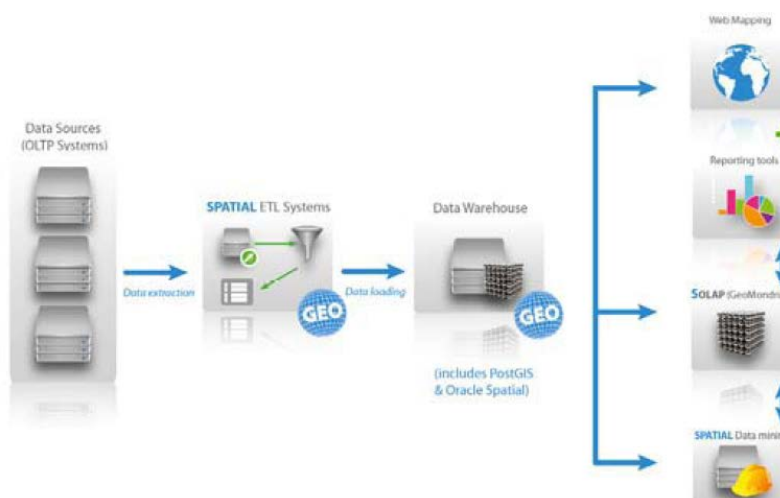


Figura 5 - Infraestrutura com Geo extensões. Fonte (Klaus Bohm, 2011)

As aplicações entre o BI e o cliente RIA foram baseadas em XMLA. O conceito de RIA inclui uma aproximação baseada em Widgets que são tratados como objetos independentes, podendo estes objetos serem facilmente integrados com o RIA. A comunicação entre widgets é baseada em eventos. Esses eventos são registados num DataManager central com triggers de eventos caso sejam manipulados dados. Os widgets foram categorizados em diferentes tipos:

- Widgets de visualização (map widgets)
- Widgets de fornecimento de dados (BI Widgets)
- Widgets de relatórios (widgets de legendas)
- Widgets de navegação (barra temporal)

O Widget de BI é utilizado para fornecer o GeoMondrian com dados utilizando a interface XMLA. Os dados fornecidos pelo BI são então guardados no DataManager que inclui um array de diferentes DataLayers. Cada DataLayer guarda os dados que deveriam ser visualizados bem como os parâmetros com que esses dados deveriam ser visualizados (técnicas de mapeamento por instancia, polígonos, etc).



Figura 6 - Arquitetura de Widgets Fonte (Klaus Bohm, 2011)

Depois de um novo DataLayer ser criado pelo widget de BI, o DataManager desplota um evento para a criação de uma nova layer de visualização e os widgets registados reagem com o evento. Por exemplo, se uma nova layer de dados é criada, é desplotado um evento e o map widget vai renderizar o mapa e o legend widget vai adicionar uma descrição aos dados mapeados. Todos os widgets mapeam os dados no mesmo DataManager.

Marco Tereso e Jorge Bernardino (2011) realizaram um estudo sobre ferramentas Open Source de BI para pequenas e médias empresas. Foi realizado um estudo sobre as ferramentas SpagoBI, OpenI, Pentaho, JasperSoft, Palo e Vanilla. Como resultado do estudo foi proposta uma tabela com a comparação entre as diversas ferramentas bem como uma análise detalhada de cada ferramenta. SpagoBI foi considerado um dos pacotes mais completos em que a sua versão única (community) contém mais funcionalidades que a grande maioria das versões (Enterprise) dos restantes pacotes. O ponto fraco da ferramenta OpenI tem a ver com o facto de não possuir métodos próprios de ETL e a impossibilidade de exportar dados. O Pentaho foi considerado uma ferramenta consistente e bastante completa a par do SpagoBI, a única diferença entre elas é o facto de o Pentaho não possuir suporte de localização GEO/GIS. A ferramenta JasperSoft carece de processos de DataMining e Key Performance Indicators (KPI) mas permite consultas ad-hoc e gestão de localização GEO/GIS. A mais-valia da ferramenta Palo é o facto de permitir consultas ad-hoc e de se poder adicionar ao Microsoft Excel e Open Office Calc. Vanilla é muito semelhante ao Pentaho mas possui uma versão para dispositivos moveis.

Funcionalidades	Ferramentas Open Source Business Intelligence					
	SpagoBI	OpenI	Pentaho	JasperSoft	Palo	Vanilla
Relatórios	✓	✓	✓	✓	✓	✓
Gráficos	✓	✓	✓	✓	✓	✓
Dashboards	✓	✓	✓	✓	✓	✓
OLAP	✓	✓	✓	✓	✓	✓
ETL	✓	✗	✓	✓	✓	✓
Data Mining	✓	✓	✓	✗	✗	✓
KPIs	✓	✗	✓	✗	✗	✓
Exportação de dados	✓	✗	✓	✓	✓	✓
GEO/GIS	✓	✗	✗	✓	✗	✗
Consultas <i>ad-hoc</i>	✓	✗	✓	✓	✓	✓
Características						
Licença GNU GPL	✓	✓	✓	✓	✓	✓
Versão Única	✓	✗	✗	✗	✓	✓
Sistema Operativo						
Linux	✓	✓	✓	✓	✓	✓
Windows	✓	✓	✓	✓	✓	✓
Unix	✓	✗	✓	✗	✗	✓

Figura 7 - Comparação de Ferramentas Open Source. Fonte (Bernardino, 2011)

Wang Lianbo, Ji Guangrong, Zheng Haiyong (2010) realizaram um estudo denominado Management Information System for Liaohe River Estuary Wetland Based on Open Source WebGIS, o qual pretendeu monitorizar dados ambientais de um estuário de um parque de reserva natural Chinês. A arquitetura escolhida baseou-se num sistema WebGIS assente no Sistema Operativo FreeBSD com o serviço de Apache para receber os pedidos HTTP (Figura 8). Esses pedidos referem-se ao serviço WMS que são transferidos para o servidor GIS que por sua vez fornece o serviço de dados acedendo à base de dados. A base de dados foi dividida em duas partes: atributos e dados espaciais. Os dados espaciais foram armazenados em dois formatos: Os dados vetoriais foram armazenados com os atributos e raster data foram armazenados em formato de ficheiro para permitir alto desempenho. Foi utilizado AJAX para construir a interface de cliente.

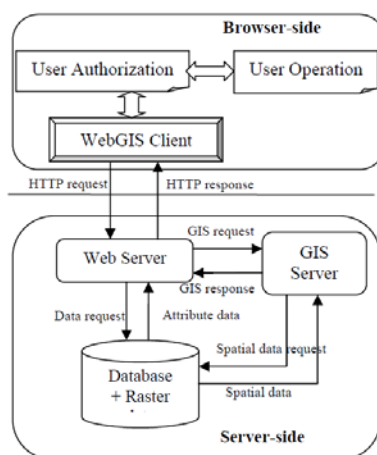


Figura 8- Arquitetura do Sistema. Fonte (Wang Lianbo, 2010)

Foi configurado o MapServer com GDAL/OGR para permitir uma variedade de formatos de dados rasters (GDAL) e vectoriais (OGR). A base de dados foi implementada em PostgreSQL com PROJ.4 para projetar cartas com WFS e WCS de forma a suportar vários standards da Open Geospatial Consortium. O núcleo do MapServer é o ficheiro Mapfile, que permite indicar ao Mapserver de onde vem os dados, quantos objetos são desenhados no mapa e com que tipo e estilo. O MapServer pode ser usado através de programação CGI ou programação recorrendo a MapScript, sendo esta biblioteca uma interface API para linguagens tais como PHP, Python, Perl, Ruby, Java e .NET.

Foi escolhida a solução de PHP e MapScript para construir a informação ambiental em que o processo passa pelos seguintes passos:

- a) Utilizadores enviam pedido HTTP ao interagirem com o site;
- b) O PHP envia os parâmetros recebidos para as funções API do MapScript;
- c) A função do API MapScript faz uma consulta aos dados espaciais de acordo com os parâmetros recebidos e a configuração do Mapfile;
- d) Com os dados devolvidos a API MapScript cria os mapas e disponibiliza outros serviços GIS;
- e) Os mapas e serviços GIS são enviados de volta para o browser do utilizador.

De acordo com o estudo, existem quatro formas de armazenar os atributos e dados espaciais:

- 1) Um sistema misto com base de dados RDBMS (Relational Database Management System) para os atributos e ficheiros de shapes tipo ESRI's para os dados espaciais;
- 2) Extended RDBMS onde a base de dados armazena atributos e dados espaciais;
- 3) RDBMS é um motor de base de dados espacial tal como o ArcSDE que trabalha como um canal de input e output de dados;
- 4) Object-Relational DBMS, tal como PostGIS que é baseado em tecnologia orientada a objetos para incrementar os tipos de dados espaciais e funções de operação.

Foi escolhido o quarto método onde os dados de atributos estão armazenados no PostgreSQL e os dados vetoriais espaciais estão na extensão PostGIS. Desta forma o PostGIS atua como uma base de dados espacial de backend ao PostgreSQL.

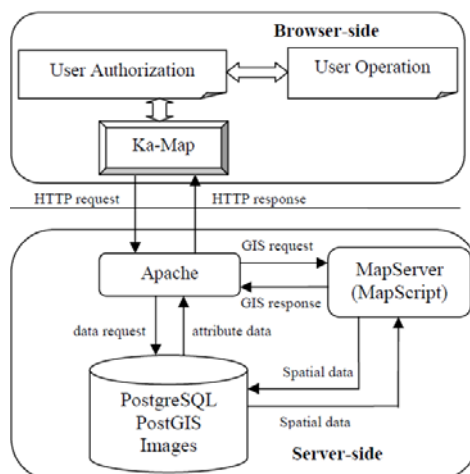


Figura 9 - Aplicações Open Source Escolhidas. Fonte (Wang Lianbo, 2010)

A tecnologia AJAX foi escolhida para criar uma aplicação interativa (como o Google Maps). Para tal o Ka-Map foi usado de forma a programar um núcleo de renderização rápido e contínuo tal como no Google Maps. Estando do lado do cliente o Ka-Map constrói um widget de mapa leve para suportar a interface através de JavaScript e HTML (Wang Lianbo, 2010).

LI Guangyu, ZHOU Kefa¹, WANG Jinlin, SUN Li, WANG Qianfeng, QIN Yanfang (2010) realizaram um estudo intitulado Implementation of Publication System of Geological Data Using Open Source GIS com o objetivo de construir um sistema integrado de gestão, publicação e análise de múltiplos objetos geo baseados numa topologia composta por múltiplas bases de dados. O armazenamento de dados foi constituído por uma base de dados espacial e uma base de dados com metadados espaciais. A BD espacial foi usada para manter múltiplas origens de dados geológicos que incluem os dados espaciais e seus atributos. A BD de metadados foi utilizada para organizar, guardar e gerir metadata com o intuito de realizar um serviço de diretório com dados de metadados espaciais. O sistema foi dividido em três camadas: representação de dados, transmissão e armazenamento de dados. A camada de transmissão é o core do WebGIS, é responsável pelos pedidos e respostas dos clientes e das bases de dados. Caso os pedidos estejam em cache são disponibilizados imediatamente, caso contrário os servidores de aplicações interagem com a camada de dados. A cama de representação é responsável pela interação com os clientes.

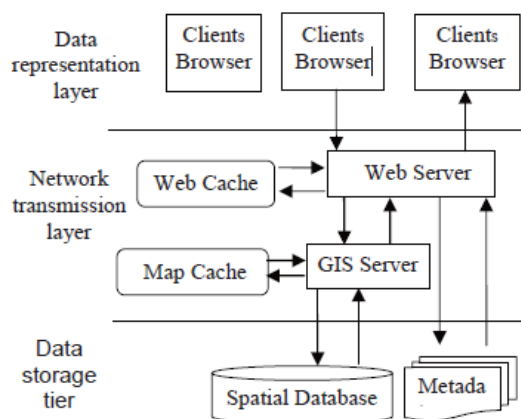


Figura 10- Arquitetura do Sistema. Fonte (LI Guangyu, 2010)

Os dados básicos armazenados na BD são a divisão de mapas, estradas, sistemas de rega e modelos de elevação (DEM). Os dados geológicos foram divididos em conjuntos de elementos básicos e conjuntos de elementos integrados. O sistema foi dividido em vários módulos:

- a) Backstage, para fazer a gestão de utilizadores, permissões, logs, backups de dados, etc;
- b) Gestão de metadados para aquisição de dados metadados, gestão da BD espacial e publicação e gestão de dados espaciais;
- c) Publicação e partilha de dados espaciais em que este módulo disponibiliza funções nos mapas tais como aproximar, afastar, deslocar, controlo de temas e vista “eagle eye”;
- d) Análise de dados espaciais, este módulo é o core do sistema e foi dividido em duas partes:
 - a. Funções de análise gerais
 - b. Funções de análise de estatísticas
- e) Módulo de output de dados, impressões, download de formulários e geração de documentação onde o utilizador pode escolher diferentes formatos de output.

O sistema foi desenvolvido utilizando o seguinte software: PostgreSQL/PostGIS 8.4, Tomcat 6.0, GeoServer 2.0.0, TileCache e Openlayers. O PostGIS permite aos utilizadores um standard de serviços WebGIS, permitindo inserção, atualização e remoção de dados geográficos de

acordo com a norma WFS. O Geoserver permite a possibilidade de publicação de mapas/imagens usando WMS e dados reais usando WFS. OpenLayers é um cliente AJAX para permitir o overlaying de vários serviços de mapas suportando vários formatos de dados tais como XML, GML, GeoJSON, GeoRSS, JSON, KML, WFS, etc. O desenvolvimento do *backstage* foi baseado no GeoServer que faz as conexões ao PostGIS, o desenho de estilo de dados com XML baseado em SLD (Styles Layer Descriptor), onde diferentes estilos podem ser aplicados a diferentes layers. O *frontpage* foi desenhado recorrendo a DIV's e CSS realizadas em Java com JSP.NET e Servlet e as funções de navegação no mapa utilizando a API OpenLayers. Na figura 11 e figura 12, são apresentados excertos de código javascript utilizando a biblioteca OpenLayers, para, respetivamente, a publicação de mapas e programação de resposta a um evento de clique sobre um interface.

```
<script src="Scripts/lib/OpenLayers.js"></script>
<script>
function init() {
  var map = new OpenLayers.Map("map-id");
  var wms = new OpenLayers.Layer.WMS(
    "alteration",
    "http://127.0.0.1:8080/geoserver/wms",
    {layers: "xxxia:alteration"});
  var wfs = new OpenLayers.Layer.Vector("dike", {
    strategies: [new OpenLayers.Strategy.BBOX()],
    protocol: new OpenLayers.Protocol.WFS({
      url: "http://127.0.0.1:8080/geoserver/wfs",
      featureType: "dike", featurePrefix: "xxxia",
      featureNS: "http://www.esrcca.com/gis",
      geometryName: 'the_geom'}}));
  map.addLayers([wms,wfs]);
  map.addControl(new OpenLayers.Control.PanZoomBar({
    position: new OpenLayers.Pixel(2, 15) }));
  map.addControl(new
    OpenLayers.Control.LayerSwitcher({'div':OpenLayers.Util
    getElement('layerswitcher')}));
  map.zoomToMaxExtent(new OpenLayers.Bounds(-
    124.731422, 24.955967,-66.969849, 49.371735)); }
</script>
```

Figura 11 - Exemplo de Código para publicação de Mapas usando OpenLayers.

Fonte (LI Guangyu, 2010)

```
map.events.register('click', map, function (e) {  
    var params = {REQUEST: "GetFeatureInfo",  
        EXCEPTIONS: "application/vnd.ogc.se_xml",  
        BBOX: map.getExtent().toBBOX(),  
        X: e.xy.x, Y: e.xy.y,  
        INFO_FORMAT: 'text/html',  
        QUERY_LAYERS: map.layers[0].params.LAYERS,  
        FEATURE_COUNT: 50,  
        Layers: xxxia:alteration',  
        Styles: '',  
        Srs: 'EPSG:4326',  
        WIDTH: map.size.w,  
        HEIGHT: map.size.h,  
        format: format};  
    OpenLayers.loadURL("http://127.0.0.1:8080/geoserver/wms",  
        params, this, setHTML, setHTML);  
    OpenLayers.Event.stop(e); }); }
```

Figura 12 - Demonstração de Código de Event Click num Mapa. Fonte (LI Guangyu, 2010)

2.6 Reflexão Crítica

Todos os casos de estudo analisados implementaram soluções baseadas em software Open Source. As arquiteturas procuraram sempre respeitar as normas da OGC. PostgreSQL com extensão PostGis foram utilizados em todos os casos de estudo. Alguns optaram pela suite Pentaho completa (GeoKettle, GeoMondrian e SolapLayer), enquanto que outros optaram por efetuar a preparação do ETL com SpagoBI ou OpenL sendo a configuração da componente georeferenciada realizada usando ferramentas independentes (GeoServer, Mapserver, OpenLayers) tendo toda a programação de integração das diversas plataformas ter sido realizada de raiz. A principal dificuldade referenciada pela maioria dos autores foi a integração e comunicação entre plataformas e a programação das interfaces cliente.

3 Caso de Estudo

3.1 Introdução

No âmbito deste trabalho de projeto, foram utilizados dados relativos a um estudo baseado num inquérito de saúde realizado pela Escola Superior de Saúde do Instituto Politécnico de Viana do Castelo e o Observatório de Saúde do Gabinete de Cidade Saudável. No âmbito do presente trabalho de projeto, foram utilizados dados relativos a um estudo epidemiológico relativo ao perfil de saúde da população de um concelho de Portugal, realizado pela Escola Superior de Saúde do Instituto Politécnico de Viana do Castelo em articulação com o Observatório de Saúde do Gabinete de Cidade Saudável. O projeto foi coordenado pelos docentes Carlos Subtil, Clara Araújo, Luís Graça e Maria Aurora Pereira da Escola Superior de Saúde do IPVC. Um dos objetivos do estudo visou georreferenciar através de ferramentas SIG a caracterização de várias doenças crónicas, saúde mental, cuidados preventivos e caracterização sociodemográfica de um concelho de Portugal. Foram utilizados como métodos, a observação, a entrevista, o questionário, a escala de medida, a avaliação física e a avaliação psicológica. Após o tratamento dos dados a amostra foi reduzida de 1519 para 1517 inquéritos. Os mesmos foram descaracterizados de forma a não violar a privacidade das pessoas inquiridas. O inquérito não careceu de validação pelo que a informação recolhida só foi utilizada no contexto de trabalho como dados de teste para demonstrar a capacidade deste tipo de ferramentas.

3.2 Contexto de aplicabilidade

Quantas pessoas do concelho estudado inquirido sofrem de diabetes no ano de 2012? E destas quantas são do sexo masculino ou feminino, ou ainda além de sofrer de diabetes quantas também tem problemas cardiovasculares?

No universo de pessoas inquiridas, quantas sofrem de stress? E destas quantas trabalham por conta própria ou por conta de outrem, que grau académico possuem essas pessoas? Todas as repostas a estas questões estariam mais enriquecidas se tivessem associada uma componente geográfica. De forma a responder a estas questões foi decidido desenvolver o projeto recorrendo a ferramentas com capacidade de integrar a componente geoespacial em todas as partes da arquitetura BI. Os sistemas de informação geográfica, através da inclusão da componente espacial dos dados, oferecem capacidades para visualizar, compreender, inquirir, interpretar dados, revelando relações, padrões e tendências espaciais.

3.3 Arquitetura Geral e Ferramentas Utilizadas

Na figura 13 podemos visualizar de forma genérica as fontes de dados utilizadas para alimentar o sistema de apoio à decisão. Foram analisadas diversas arquiteturas aquando do estudo de trabalhos relacionados, e optou-se pela arquitetura geral, com fontes de dados diversas, o GeoKettle para extrair e carregar dados para a base de dados PostgreSQL com extensão PostGis, o GeoMondrian para permitir a integração de objetos espaciais na estrutura de cubo de dados OLAP, o JPivot para a renderização dos cubos e o OpenLayers para a visualização de mapas do lado do cliente.

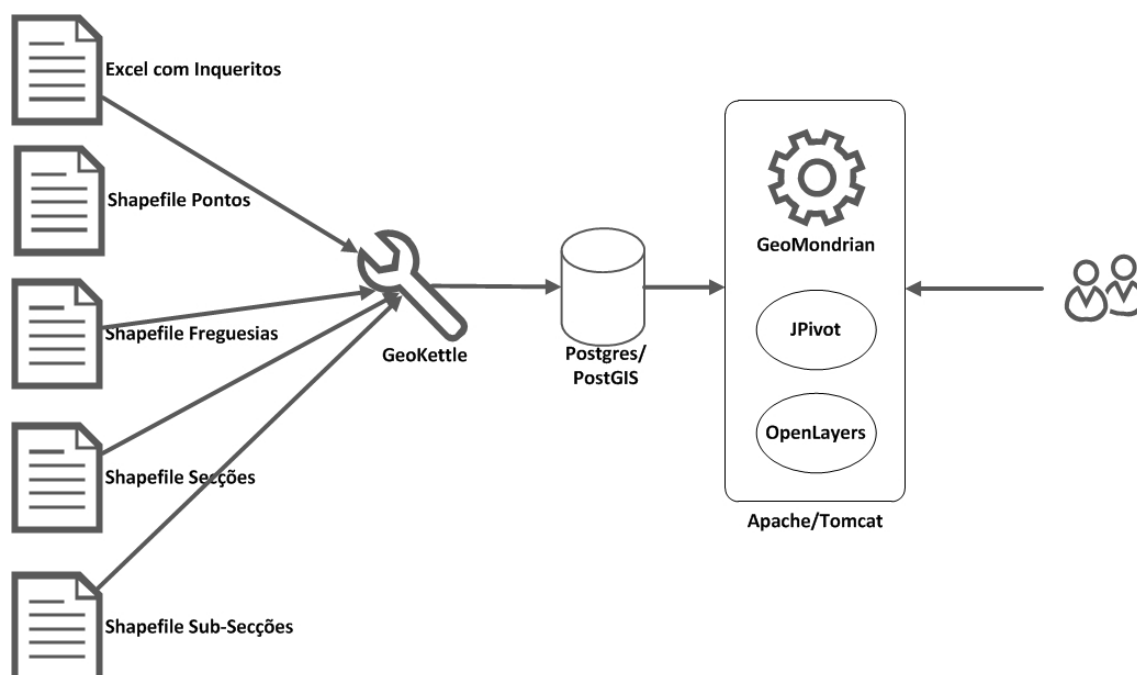


Figura 13 – ETL, OLAP e Ferramentas de BI.

Em termos de execução, optou-se por criar uma máquina virtual recorrendo ao software VMWare Workstation versão 8.0.3 build-703057. O sistema operativo anfitrião escolhido foi Windows 7 versão Ultimate 32 bits com service pack 1. A máquina física era constituída pelas seguintes características, processador Intel i3 M350 2.27GHz, 8GB de memória RAM dos quais foram alocados 4GB de memória RAM para a máquina virtual (dos quais 3 utilizáveis devido à versão 32bits) e dois processadores com dois cores por processador, o disco virtual foi criado com 60GB o que se revelou mais do que suficiente ao longo do projeto. Foram utilizadas ao longo do projeto as seguintes ferramentas:

- **Excel 2010:** Folha de Cálculo do Microsoft Office. A utilização desta ferramenta centrou-se no requisito das fontes de dados.
- **QuantumGis:** Versão 1.8.0 “Lisboa” - O Quantum GIS (QGIS) é um sistema livre de informação geográfica (SIG) multiplataforma que suporta formatos vetoriais, raster, e acesso a bases de dados. O QGIS permite procurar, editar e criar formatos ESRI shapefiles, dados

espaciais em PostgreSQL/PostGIS, vectores e rasters GRASS, ou ainda GeoTiff. O QGIS suporta ainda extensões e acesso a módulos do GRASS, permitindo visualizar mapas do GRASS em simultâneo com dados SIG noutros formatos. (Livre, s.d.).

- **Geokettle** Versão 3.2: é uma ferramenta espacial de ETL, orientada por metadados dedicada à integração de diferentes fontes de dados espaciais para a construção e atualização de data warehouses geoespaciais. O GeoKettle é um software que deriva do kettle (que faz parte da suite Pentaho) ao qual foram adicionadas extensões espaciais. Tem a capacidade de executar transformações diretamente sem necessidade de escrita de código, suporta 37 motores de bases de dados, escreve e lê diferentes tipos de formatos de ficheiros (excel, access, DBF, XML). GeoKettle permite diversas operações tais como:
 - o extração de dados de diferentes fontes de dados;
 - o transformação de dados, a fim de corrigir erros ou fazer limpeza de dados, alterar a estrutura de dados, torná-los compatíveis com os padrões definidos;
 - o e o carregamento de dados transformados numa base de dados (DBMS) em OLTP ou OLAP / SOLAP, arquivo GIS ou Serviço Web Geoespacial.

- **PostgreSQL** versão 9.1: é um sistema gestor de bases de dados objeto relacional, desenvolvido como projeto de código aberto, que tem como principais características:
 - o Consultas complexas;
 - o Chaves estrangeiras;
 - o Integridade transacional;
 - o Controle de concorrência multi-versão;
 - o Suporte ao modelo híbrido objeto-relacional;
 - o Triggers
 - o Views;

- o Linguagem procedimental em várias linguagens (PL/pgSQL, PL/Python, PL/Java, PL/Perl) para a criação de procedimentos armazenados;
 - o Indexação por texto;
 - o Estrutura para guardar dados georeferenciados recorrendo à extensão PostGIS.
- **PostGis** versão 2.0: PostGIS é uma extensão do sistema de base de dados PostgreSQL, que permite que objetos georreferenciados possam ser armazenados em bases de dados. PostGIS inclui o suporte para índices espaciais GiST-baseados em R-Tree e funções para análise e processamento de objetos georeferenciados. (Webgis, s.d.).
- **Geomondrian** versão 1.0: é um Servidor Open Source de Processamento Analítico Online Espacial. É uma versão a que foram acrescentadas funções de processamento espacial a partir do Mondrian Pentaho Analysis Services. GeoMondrian é uma implementação de um servidor SOLAP. Permite a integração consistente de objetos espaciais na estrutura do cubo de dados OLAP. Fazendo uma analogia simples, GeoMondrian traz para o servidor OLAP Mondrian o que o PostGIS traz para o sistema de gestão de bases de dados PostgreSQL. Implementa um tipo de dados geométricos nativo e fornece extensões espaciais para a linguagem de consulta MDX, permitindo a incorporação de recursos de análise espaciais em consultas analíticas, (Spatialytics, s.d.). Através destas extensões geoespaciais a linguagem de consulta MDX fornece muitas mais possibilidades, tais como:
 - o construtores geométricos in-line;
 - o filtros com base em predicados topológicos;
 - o membros e medidas espacialmente calculados;
 - o cálculos com base em atributos escalares derivados de características espaciais;
- **GeoWorkBench** versão 3.1.1 Revision 12687: é uma aplicação que deriva do Workbench Mondrian Schema. Consiste numa aplicação que permite criar e testar visualmente esquemas cubo SOLAP no GeoMondrian. O motor do GeoMondrian processa os pedidos MDX com

os esquemas (Relational OLAP) ROLAP que são criados e publicados para o GeoMondrian através de arquivos metadados em XML. Estes arquivos são criados usando uma estrutura especificamente utilizada pelo motor do GeoMondrian (Pentaho, s.d.).

- **Apache/Tomcat** versão 7.0: é um servidor web Java, mais especificamente, um container de servlets. Desenvolvido pela Apache Software Foundation, é distribuído como software livre dentro do conceituado projeto Apache Jakarta (Wikipedia, s.d.).
- **JPivot** versão 1.8.0: JPivot é uma biblioteca JSP que permite a renderização de cubos OLAP permitindo que os utilizadores finais possam navegar ao longo do cubo efetuando operações tais como slice, dice, drill-down ou roll-up. Utiliza o servidor OLAP Mondrian ou Geomondrian.
- **Openlayers** versão 2.12: é uma biblioteca escrita em JavaScript que permite a visualização e colocação de objetos em mapas. A sua utilização é possível em todos os browsers mais recentes sem dependências do lado do servidor. Implementa uma API em Javascript que constrói paginas web interativas com objetos geográficos.
- **Ext JS** versão 3.4.0: é uma biblioteca que permite ao programador construir aplicações para a web usando JavaScript com o standarts Web.

3.4 Características das Fontes de Informação

Para a realização deste projeto foram disponibilizadas várias fontes de dados, que incluíram dados sobre os inquéritos e sua localização geográfica e sobre a divisão do espaço geográfico proveniente do Instituto Nacional de Estatística (INE). Um primeiro ficheiro no formato Excel (Figura 14) continha as respostas aos 1517 inquéritos realizados nas freguesias do concelho em estudo. O inquérito é composto de 42 items, dos quais 11 caracterizam o entrevistado e os restantes 31 questionam o entrevistado acerca de diversas doenças.

As questões que caracterizam do ponto de vista socio demograficamente o entrevistado são:

- FRG11 – Número que representa a freguesia baseado nos dados do INE censos 2011.
- SEC11 - Número que representa a secção dentro da freguesia baseado nos dados do INE censos 2011.
- codse11 - Número concatenado com a freguesia mais a secção baseado nos dados do INE censos 2011.
- Freguesia (a0001) – Nome da freguesia
- Inquérito Nº (a0001a) – Número do inquérito
- Morada (a0006)
- Sexo (a0100d)
- A sua idade é (a0103b)
- Qual é o [seu] estado civil legal? (a0104)
- Qual o nível de ensino mais elevado frequentado? (a0106)
- Das seguintes categorias, qual a que melhor descreve a [sua] ocupação principal, nas duas últimas semanas? (a0108a)

As questões relacionadas com saúde:

- O(a) Sr(a) tem ou já teve diabetes (açúcar no sangue)? (a0301)
- [O(a) Sr(a)] tem ou já teve asma? (a0310)
- [O(a) Sr(a)] tem ou já teve tensão arterial alta (hipertensão arterial)? (a0316)
- [O(a) Sr(a)] tem ou já teve dor crónica (dor constante ou repetitiva durante, pelo menos, três meses)? (a0321)
- [O(a) Sr(a)] tem ou já teve doença reumática (osteoartrose, tendinite)? (a0324a)
- [O(a) Sr(a)] tem ou já teve Osteoporose? (a0324b)
- [O(a) Sr(a)] tem ou já teve Glaucoma? (a0324c)
- [O(a) Sr(a)] tem ou já teve Retinopatia (doença da retina)? (a0324d)
- [O(a) Sr(a)] tem ou já teve tumor maligno ou Cancro? (a0324e)
- [O(a) Sr(a)] tem ou já teve pedra nos rins? (a0324f)

- [O(a) Sr(a)] tem ou já teve insuficiência renal? (a0324g)
- [O(a) Sr(a)] tem ou já teve ansiedade crónica? (a0324h)
- [O(a) Sr(a)] tem ou já teve ferida crónica (úlceras da perna, escaras)? (a0324i)
- [O(a) Sr(a)] tem ou já teve Enfisema (doença pulmonar obstrutiva crónica) bronquite crónica? (a0324j)
- [O(a) Sr(a)] tem ou já teve Avc (acidente vascular cerebral)? (a0324k)
- [O(a) Sr(a)] tem ou já teve Obesidade? (a0324l)
- [O(a) Sr(a)] tem ou já teve Depressão? (a0324m)
- [O(a) Sr(a)] tem ou já teve Enfarte de Miocárdio? (a0324n)
- [O(a) Sr(a)] tem ou já teve... (outra doença) (a0324o)
- Saúde Mental - Nas últimas 4 semanas, quanto tempo se sentiu muito nervoso? (a0701)
- Nas últimas 4 semanas, quanto tempo se sentiu tão deprimido(a) que nada o(a) animava? (a0702)
- Nas últimas 4 semanas, quanto tempo se sentiu calmo(a) e tranquilo(a)? (a0703)
- Nas últimas 4 semanas, quanto tempo se sentiu triste/desanimado(a) e em baixo/abatido(a)? (a0704)
- Nas últimas 4 semanas, quanto tempo se sentiu feliz? (a0705)
- Quando foi a última vez que [o(a) Sr(a)] se vacinou contra a gripe? (a0801)
- Quando foi a última vez que [o(a) Sr(a)] verificou (mediu) a [sua] tensão arterial? (a0803)
- Em que ano fez a última mamografia, isto é, uma radiografia ao peito? (a0805)
- Em que ano fez a última Citologia (esfregaço vaginal/teste de Papanicolau)? (a0806)
- Em que ano fez o último exame ao intestino (colonoscopia, sigmoscopia)? (a0807b)
- Em que ano fez o último exame à Próstata - toque rectal? (a0808b)
- Em que ano fez o último exame à próstata - Teste PSA (exame ao sangue)? (a0809)

“Business Intelligence” sobre informação espacial: um caso de estudo

The screenshot shows a Microsoft Excel spreadsheet titled "Base de Dados Final com tabelas-V2-2003-original [Modo de Compatibilidade] - Microsoft Excel". The ribbon includes "Ficheiro", "Base", "Inserir", "Esquema de Página", "Fórmulas", "Dados", "Rever", and "Ver". The "Dados" tab is active, showing a data table with columns L through U. The table contains 30 rows of data, each representing a record with responses to various health-related questions. The questions are: "O(a) Sr(a) tem ou já teve diabetes (açúcar no sangue)? (a0301)", "O(a) Sr(a) tem ou já teve asma? (a0310)", "O(a) Sr(a) tem ou já teve tensão arterial alta (hipertensão arterial)? (a0316)", "O(a) Sr(a) tem ou já teve... (a0317)", "O(a) Sr(a) tem ou já teve... (a0318)", "O(a) Sr(a) tem ou já teve... (a0319)", "O(a) Sr(a) tem ou já teve... (a0320)", "O(a) Sr(a) tem ou já teve... (a0321)", "O(a) Sr(a) tem ou já teve... (a0322)", "O(a) Sr(a) tem ou já teve... (a0323)", "O(a) Sr(a) tem ou já teve... (a0324)", "O(a) Sr(a) tem ou já teve... (a0325)", "O(a) Sr(a) tem ou já teve... (a0326)", "O(a) Sr(a) tem ou já teve... (a0327)", "O(a) Sr(a) tem ou já teve... (a0328)", "O(a) Sr(a) tem ou já teve... (a0329)", "O(a) Sr(a) tem ou já teve... (a0330)", "O(a) Sr(a) tem ou já teve... (a0331)", "O(a) Sr(a) tem ou já teve... (a0332)", "O(a) Sr(a) tem ou já teve... (a0333)", "O(a) Sr(a) tem ou já teve... (a0334)", "O(a) Sr(a) tem ou já teve... (a0335)", "O(a) Sr(a) tem ou já teve... (a0336)", "O(a) Sr(a) tem ou já teve... (a0337)", "O(a) Sr(a) tem ou já teve... (a0338)", "O(a) Sr(a) tem ou já teve... (a0339)", "O(a) Sr(a) tem ou já teve... (a0340)". The responses are "Sim" or "Não".

	L	M	N	O	P	Q	R	S	T	U
1	O(a) Sr(a) tem ou já teve diabetes (açúcar no sangue)? (a0301)	O(a) Sr(a) tem ou já teve asma? (a0310)	O(a) Sr(a) tem ou já teve tensão arterial alta (hipertensão arterial)? (a0316)	O(a) Sr(a) tem ou já teve... (a0317)	O(a) Sr(a) tem ou já teve... (a0318)	O(a) Sr(a) tem ou já teve... (a0319)	O(a) Sr(a) tem ou já teve... (a0320)	O(a) Sr(a) tem ou já teve... (a0321)	O(a) Sr(a) tem ou já teve... (a0322)	O(a) Sr(a) tem ou já teve... (a0323)
5	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
6	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
9	Não	Sim	Não	Não	Sim	Não	Não	Não	Não	Não
12	Não	Sim	Não	Não	Não	Não	Não	Não	Não	Não
15	Não	Não	Não	Não	Sim	Não	Não	Não	Não	Não
17	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
18	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
19	Sim	Não	Não	Não	Não	Não	Não	Não	Não	Não
20	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
22	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
24	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
25	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
29	Sim	Não	Não	Não	Não	Não	Não	Não	Não	Não
30	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
34	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
35	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
36	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
39	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
40	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
41	Sim	Não	Sim	Não	Não	Não	Não	Não	Não	Não
42	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
44	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
47	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
49	Não	Não	Sim	Não	Não	Não	Não	Não	Não	Não
50	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
51	Não	Não	Sim	Não	Não	Não	Não	Não	Não	Não
57	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
58	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
62	Não	Não	Não	Não	Não	Não	Sim	Não	Não	Não
67	Não	Não	Sim	Não	Não	Não	Não	Não	Não	Não
71	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
72	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
81	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
82	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
84	Não	Não	Sim	Não	Não	Não	Não	Não	Não	Sim
86	Não	Não	Não	Não	Não	Não	Não	Não	Não	Sim

Figura 14 - Amostra de dados fornecidos no Excel.

Além do ficheiro excel com as respostas aos 1517 inquéritos, foram também disponibilizados quatro ficheiros shapefile. O shapefile é um formato de ficheiro que contém dados geoespaciais vetoriais, podendo conter informação sobre pontos, linhas e polígonos. São usados para representar várias entidades geométricas, tais como estradas, pontes, rios, podendo cada item conter propriedades ou atributos de forma a descrever nomes, profundidade ou temperatura. Dos quatro shapefiles fornecidos, um (Figura 15) contém as coordenadas onde cada entrevista foi realizada e o número do inquérito. Este shapefile foi utilizado para no processo de ETL acrescentar em cada registo (entrevista) da tabela de factos a coordenada onde a mesma foi realizada. Os restantes três shapefiles (Figura 16, Figura 17 e Figura 18) contêm respetivamente os polígonos que delimitam cada freguesia, secção e subsecção. Apesar da fonte de dados em excel não conter o código numérico que identifica a subsecção onde a entrevista foi realizada, optou-se por exportar para o data warehouse esta informação geográfica de forma a facilitar trabalho futuro.

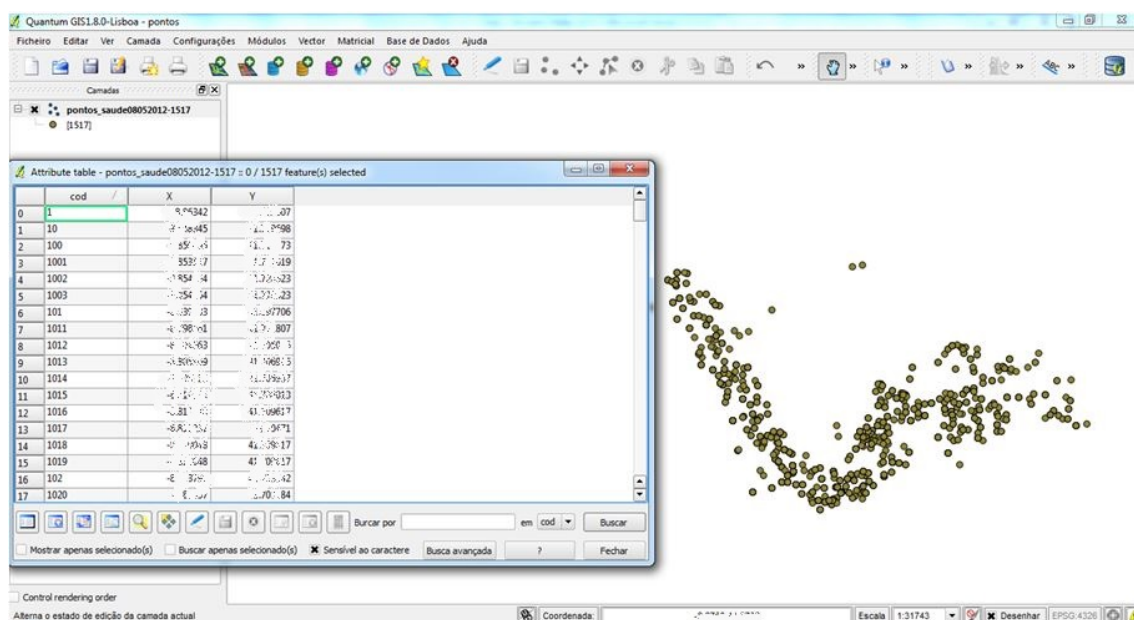


Figura 15 - Shapefile pontos.

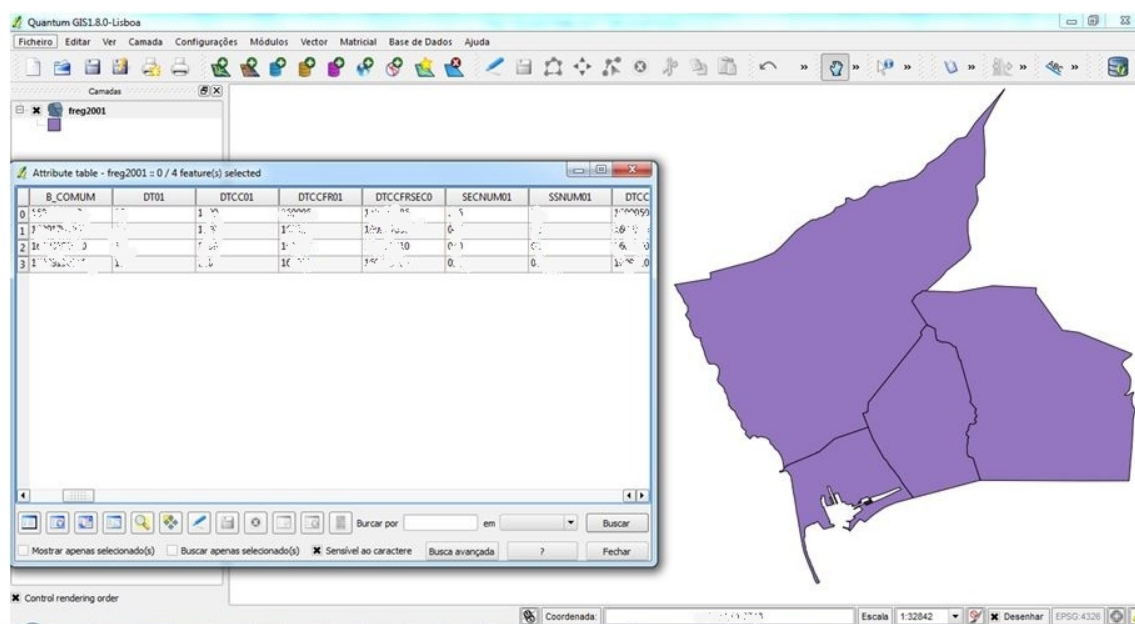


Figura 16 - Shapefile Freguesias.

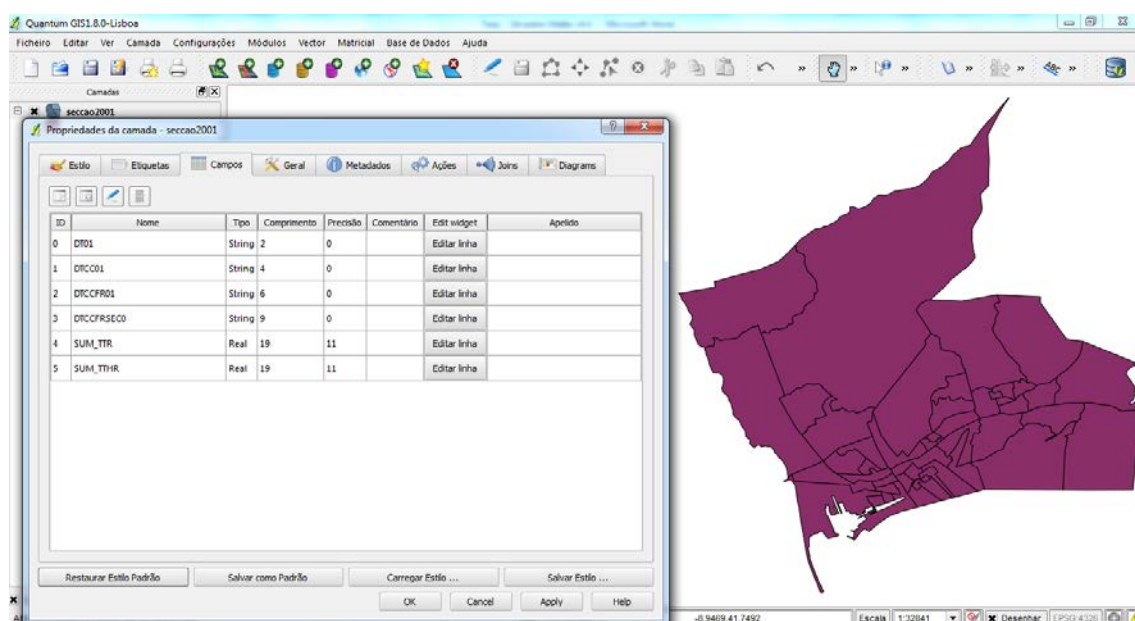


Figura 17 - Shapefile Secções.

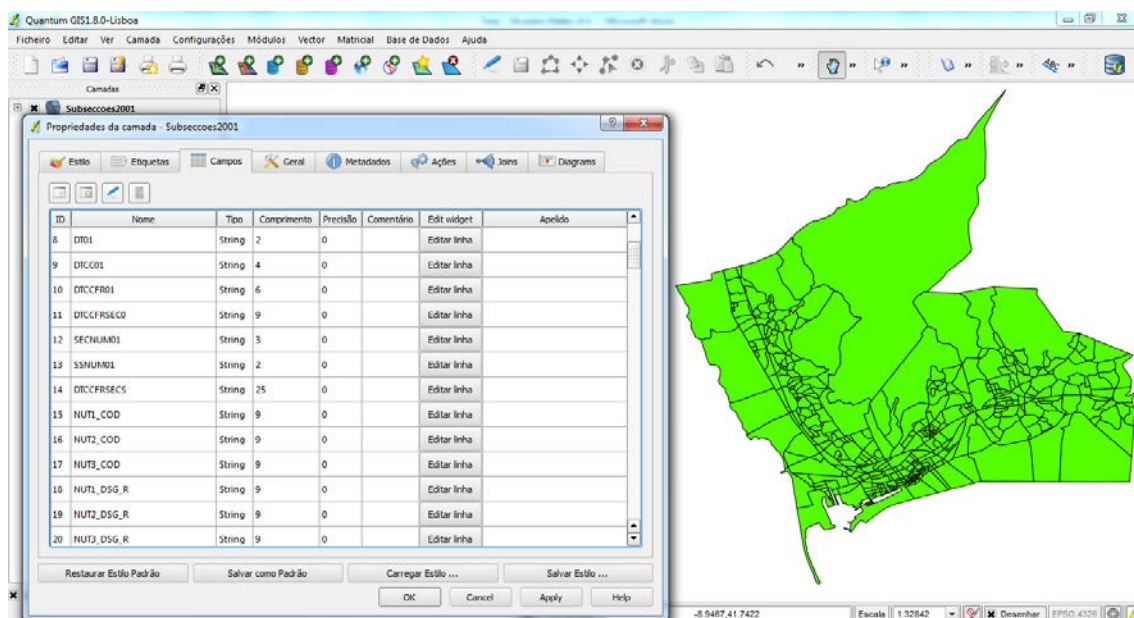


Figura 18 - Shapefile Sub Secções.

3.5 Construção do modelo multidimensional

Tipicamente, no desenho e construção do modelo multi-dimensional, uma das grandes linhas de orientação é a de Kimball [Kimball, 1996] em que o processo se inicia pela identificação das necessidades de implementação do Data Warehouse no suporte às ferramentas de Business Intelligence (apresentado na secção 2.2), pelo processo de desenho que se inicia pela definição de requisitos e análise, desenho e planeamento. Neste ponto, em termos de requisitos, o contexto deste trabalho apresenta grandes volumes de dados, a necessidade de análise georreferenciada, a necessidade da obtenção de consultas rápidas e de diferentes variáveis de análise. Em termos de

identificação das dimensões de análise, esta é conseguida através da criação de uma matriz de identificação das dimensões com base nas questões dos agentes de decisão (apresentadas na secção 3.4).

Neste sentido, na construção do modelo multidimensional, a abordagem efetuada resultou num misto de top-down e bottom-up em termos de análise de informação, no sentido de se poder analisar grandes volumes de dados, analisar as respostas às questões das fontes de dados e permitir que a ferramenta de Business Intelligence explore relacionamentos entre questões. Inicialmente foi projetado um desenho e um planeamento geral da data warehouse, mas verificou-se que a fonte de dados continha inconsistências que não poderiam ser resolvidas através do processo de ETL. Foi então facultado um novo ficheiro excel com as inconsistências resolvidas. A segunda e última versão do modelo OLAP sofreu algumas alterações. A experiência adquirida na construção do primeiro modelo OLAP fez com que a construção do segundo modelo fosse um misto de top-down/bottom-up devido às experiências de prototipagem adquiridas no primeiro planeamento. Devido à natureza dos dados facultados como resultado da natureza das fontes de informação o “processo de negócio” não poderia deixar de ser o estudo do inquérito nas suas variadíssimas variantes (doenças). O grão da data warehouse é o inquérito analisado individualmente. Foram identificadas várias dimensões, e considerando que é inquérito relacionado com o estudo de doenças, as dimensões escolhidas refletem cada doença estudada bem como os dados sócio demográficos dos indivíduos inquiridos. A figura 19 ilustra o modelo multidimensional implementado com a tabela de factos e as várias dimensões consideradas.

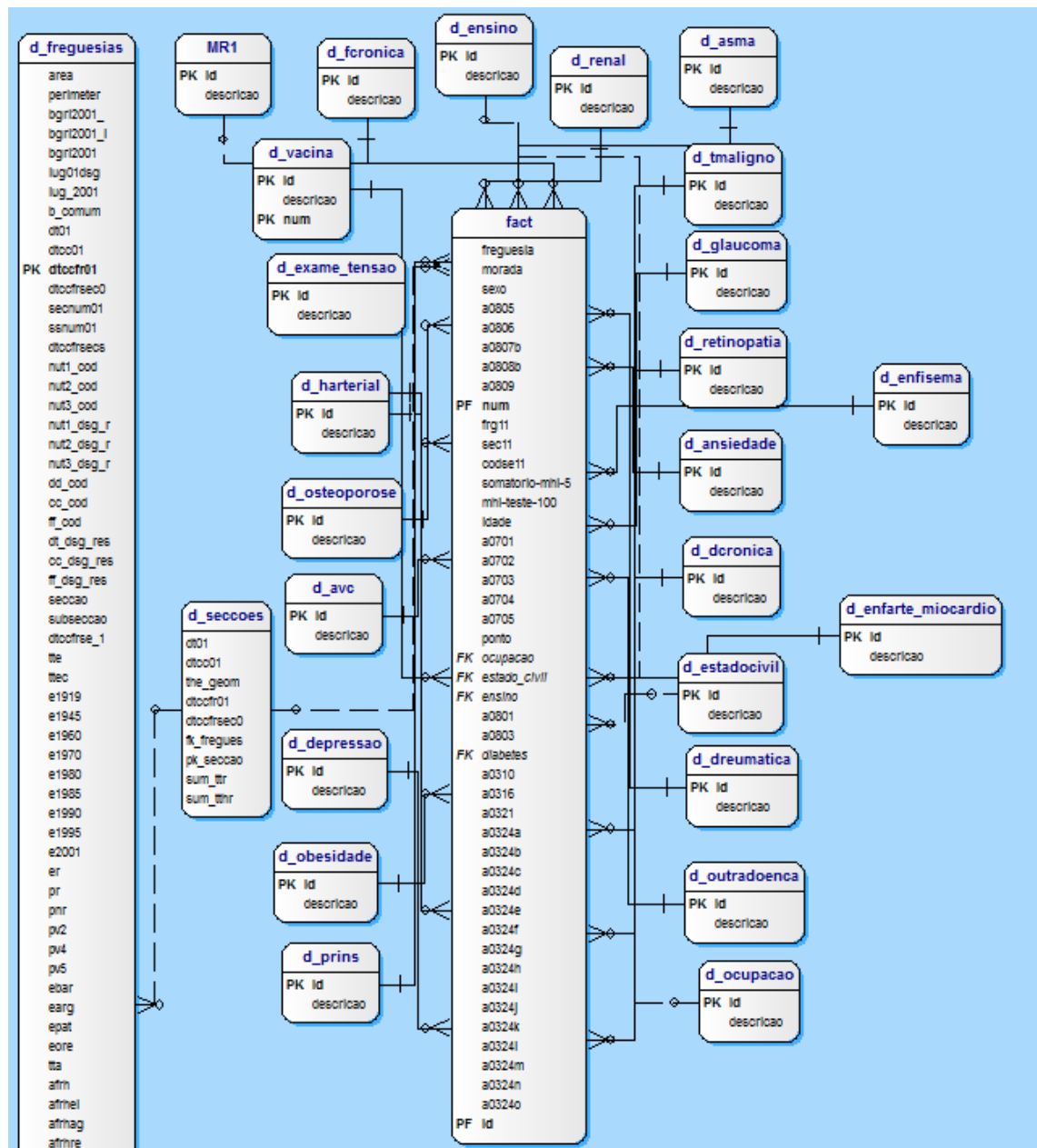


Figura 19 - Modelo Multidimensional

As dimensões identificadas foram as seguintes:

- Freguesia
- Secção
- Estado civil
- Nível de ensino
- Ocupação principal

- Diabetes
- Asma
- Hipertensão arterial
- Dor crónica
- Doença reumática
- Osteoporose
- Glaucoma
- Retinopatia
- Tumor maligno
- Pedra nos rins
- Insuficiência renal
- Ansiedade crónica
- Ferida crónica
- Enfisema
- Avc
- Obesidade
- Enfarte de Miocárdio
- Outra doença

Relativamente às medidas foram escolhidas:

- Quantos
- Nervoso nas últimas 4 semanas
- Depressão nas últimas 4 semanas
- Tranquilidade nas últimas 4 semanas
- Tristeza nas últimas 4 semanas
- Felicidade nas últimas 4 semanas

A medida “Quantos” permite contabilizar quantos indivíduos satisfazem os critérios de pesquisa, por exemplo, Quantos indivíduos residentes na freguesia 1, são casados que exercendo uma profissão tiveram um AVC? ou quantos indivíduos da Freguesia 4 divorciados são Obesos.

As restantes medidas contabilizam em dias a saúde mental do inquirido.

De forma a tornar a base de dados georreferenciada foi necessário carregar as funções de postGis no momento da sua criação. Optou-se pelo carregamento do template *template_postgis* disponível na instalação de postGis.

Este template ativa a linguagem PL/pgSQL (pois grande parte das funções GIS estão escritas nesta linguagem), carrega os objetos e definições das funções GIS que podem ser consultadas no ficheiro *postgis.sql*, o sistema completo de identificação de coordenadas EPSG também está contemplado no template, permitindo o uso de operações geométricas *ST_Transform()* disponíveis na tabela *spatial_ref_sys* que é automaticamente adicionada à base de dados criada.

3.6 O Processo de Extração, Transformação, e Carregamento

O processo de ETL foi realizado recorrendo à ferramenta GeoKettle. O primeiro passo do processo passa pela leitura do ficheiro Excel onde estão as respostas aos 1517 inquéritos realizados. A ferramenta tem quatro ambientes de trabalho:

- Spoon – Ambiente gráfico onde são realizadas as edições das transformações e jobs;
- Pan – Lançamento das transformações por linha de comandos;
- Kitchen – Lançamento dos jobs por linha de comando;
- Carte – Serviço que permite o lançamento de transformações e jobs remotamente;

O processo de ETL foi realizado recorrendo apenas ao Spoon e está dividido em seis transformações:

- Gripe_Tensao, executa a leitura dos campos do excel e copia os mesmos para uma tabela auxiliar na base de dados;

- Gripe_tensao_cont, usa a tabela auxiliar criada no passo anterior e alimenta as tabelas de dimensão *d_exame_tensao* e *d_vacina*;
- Excel-to-DW, onde são extraídos e transformados os dados que estão no Excel. É neste passo que a maioria dos dados da tabela de factos é carregada;
- Update_fact, é executada uma script sql, que mapeia as chaves estrangeiras tensão arterial e vacinação contra gripe da tabela de factos com as respetivas tabelas de dimensão.
- Pontos-to-DW, neste passo é realizada a leitura do shapefile contendo as coordenadas onde cada entrevista foi realizada
- Freg-Seccao-to-DW, esta transformação lê as shapefiles com os polígonos geométricos que delimitam as fronteiras de cada freguesia e secção.

Para que cada transformação seja executada à vez, foi criado um Job (Figura 20) que vai executando as transformações sequencialmente à medida que finalizam. A utilização de transformações revelou-se ser uma boa aposta, pois permitiu ir construindo e testando o processo de ETL em fases individuais, evitando o tempo de espera em tarefas cuja prototipagem já estava concluída e testada.

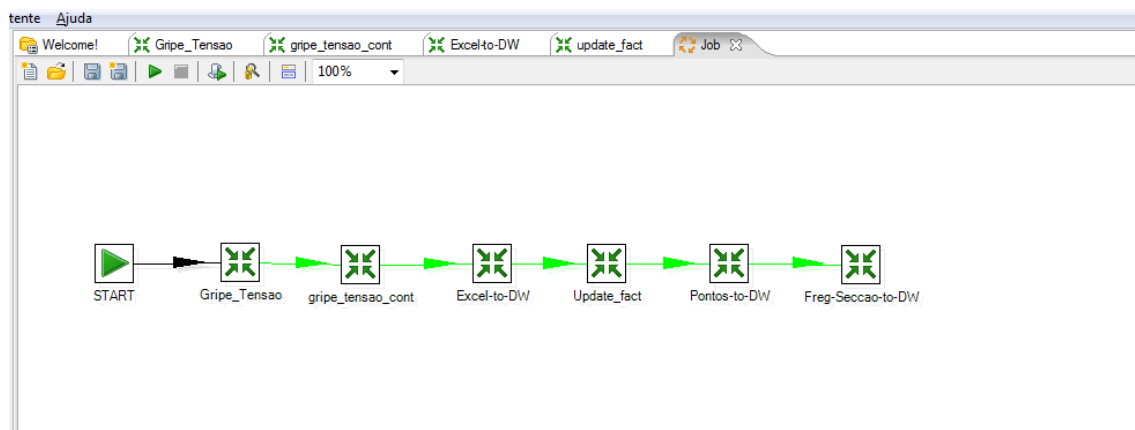


Figura 20 - Job inicial.

Nos pontos seguintes, realiza-se uma descrição com mais pormenor das transformações realizadas no processo ETL.

O step `grip_tensao_cont`, efetua algumas das seguintes tarefas (Figura 21 e Figura 22):

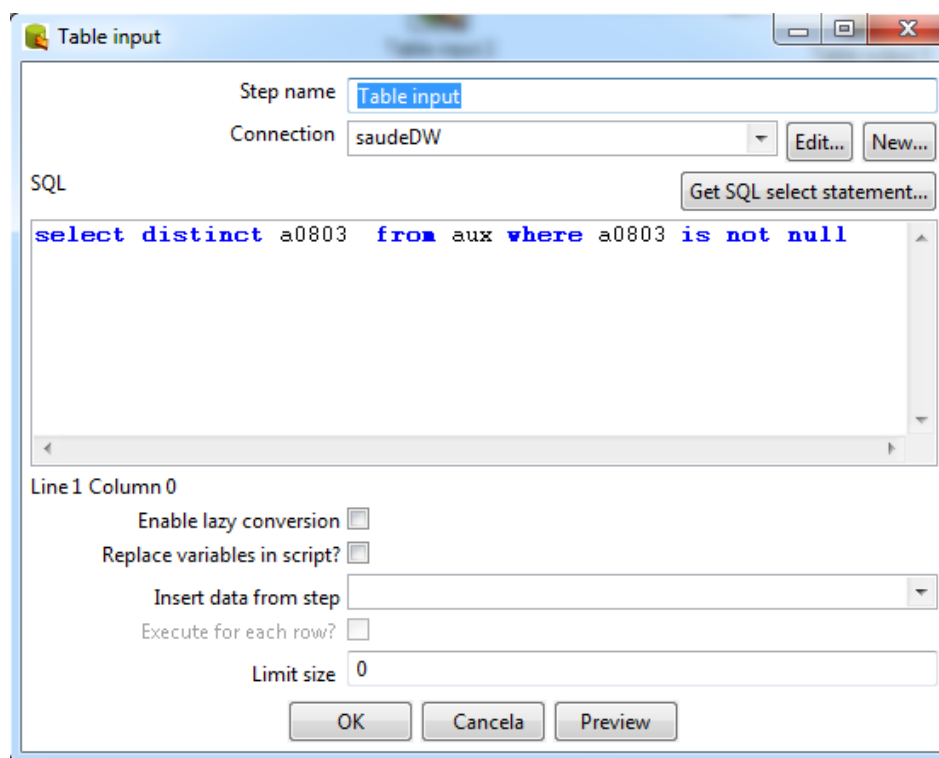


Figura 21 - Leitura distinta de a0803

São lidos distintamente os períodos de tempo da última medição da tensão arterial. Após esta leitura efetua-se a gravação dos dados na respetiva tabela de dimensão. Os restantes passos *table input 2* e *table output 3* efetuam o mesmo tipo de operações mas neste caso para a vacinação contra a gripe

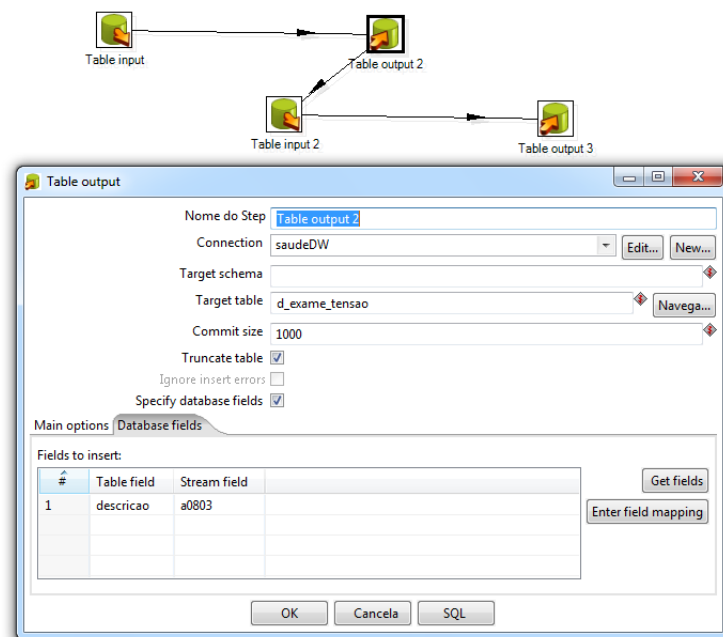


Figura 22 - Alimentação de tabelas de dimensão

De seguida passo a explicar a transformação Excel-to-DW (Figura 23). Esta transformação visa ler os dados contidos na folha de excel e executar as devidas transformações antes da gravação na datawarehousse.

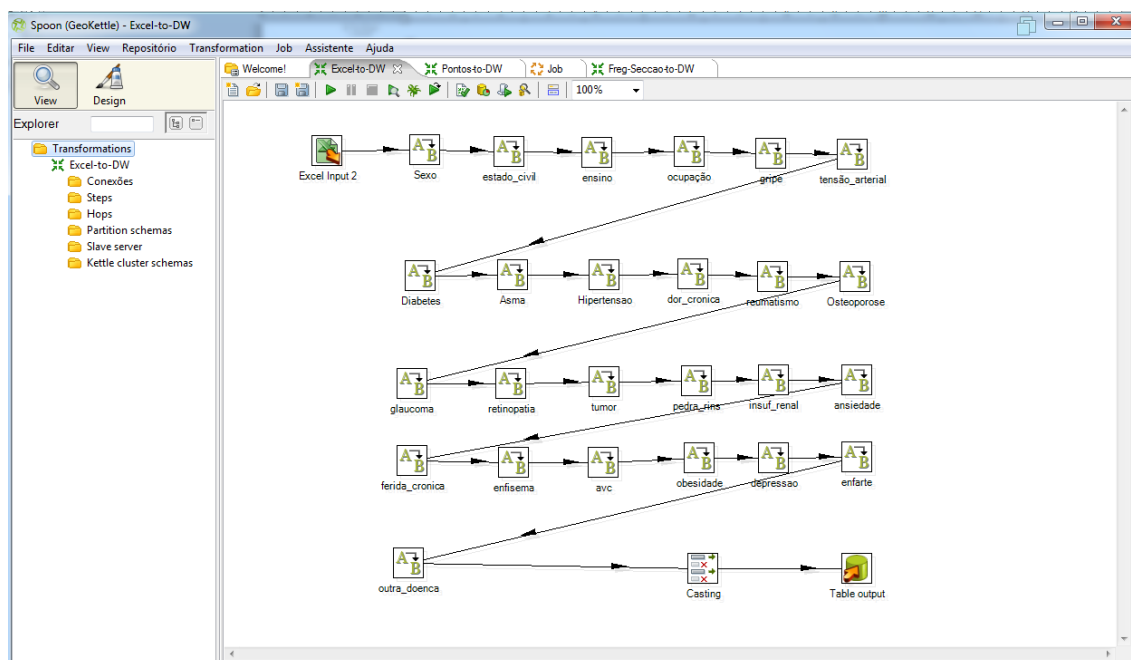


Figura 23 - Transformação Excel-to-DW.

No primeiro passo configura-se o caminho até ao ficheiro excel e escolhe-se a folha onde estão contidos os dados. Para que a transformação possa reconhecer os dados e dividi-los nos respetivos campos, é necessário executar o “get fields from header now” (Figura 24). Esta opção vai definir os nomes e tipos de dados dos campos que serão usados pelas transformações seguintes, bem como os dados (linha a linha) que estão inerentes a cada campo.

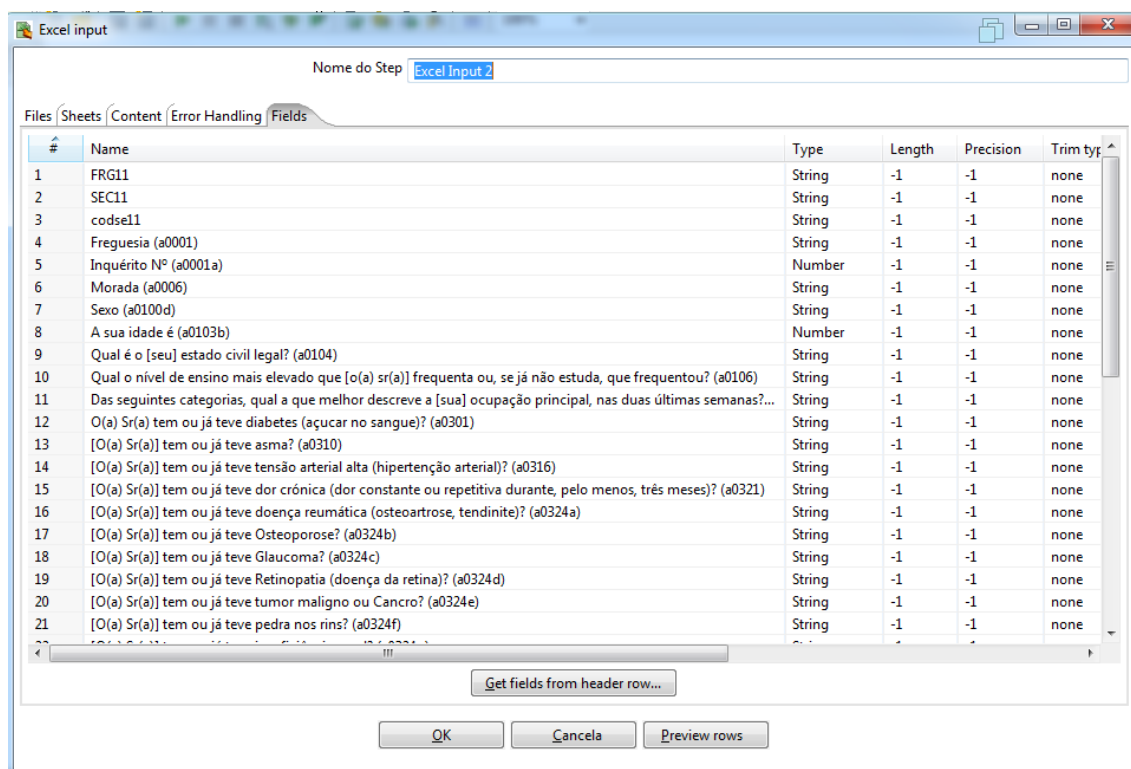
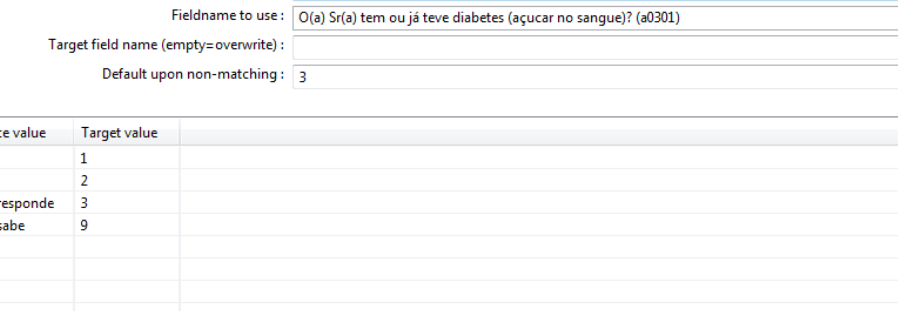


Figura 24 - Excel Input.

Após fazer a leitura dos valores do Excel, os próximos passos passam a executar a transformação dos valores lidos. A fonte de dados reflete principalmente as respostas a cada questão realizada aos inquiridos e a maior parte das respostas contem um valor restrito de hipóteses sendo, como é o caso da maioria das doenças (sim, não, não sabe, não sabe) (Figura 25) ou por exemplo no caso do estado civil (Casado(a), Casado(a) (na situação de separado legalmente), Não sabe, Solteiro(a), Viúvo(a)), Divorciado(a)) (Figura 26). De forma a evitar a escrita de dados alfanuméricos na tabela de factos e preparar a correspondência com as tabelas de dimensão, optou-se por utilizar a ferramenta *value mapper* para realizar um mapeamento das Strings devolvidas na origem por um valor numérico correspondente. Este valor

numérico está refletido numa tabela de dimensão correspondente que contém dois campos, o *id* e a sua *descrição*, por exemplo, se aquando da leitura do campo “Qual é o [seu] estado civil legal?” for devolvido a String “Casado”, o registo escrito na tabela de factos irá conter o valor 2, que por sua vez é mapeado com a tabela de dimensão *d_estado_civil* que faz a correspondência com o valor “Casado”. Nas figuras 25 e 26, apresentam-se dois exemplos de mapeamento de texto devolvido pelo excel em valores numéricos.



Value Mapper

Step name: **Diabetes**

Fieldname to use: **O(a) Sr(a) tem ou já teve diabetes (açúcar no sangue)? (a0301)**

Target field name (empty=overwrite):

Default upon non-matching: **3**

Field values:

#	Source value	Target value
1	Sim	1
2	Não	2
3	Não responde	3
4	Não sabe	9

OK Cancela

Figura 25 - Mapeamento da doença Diabetes.

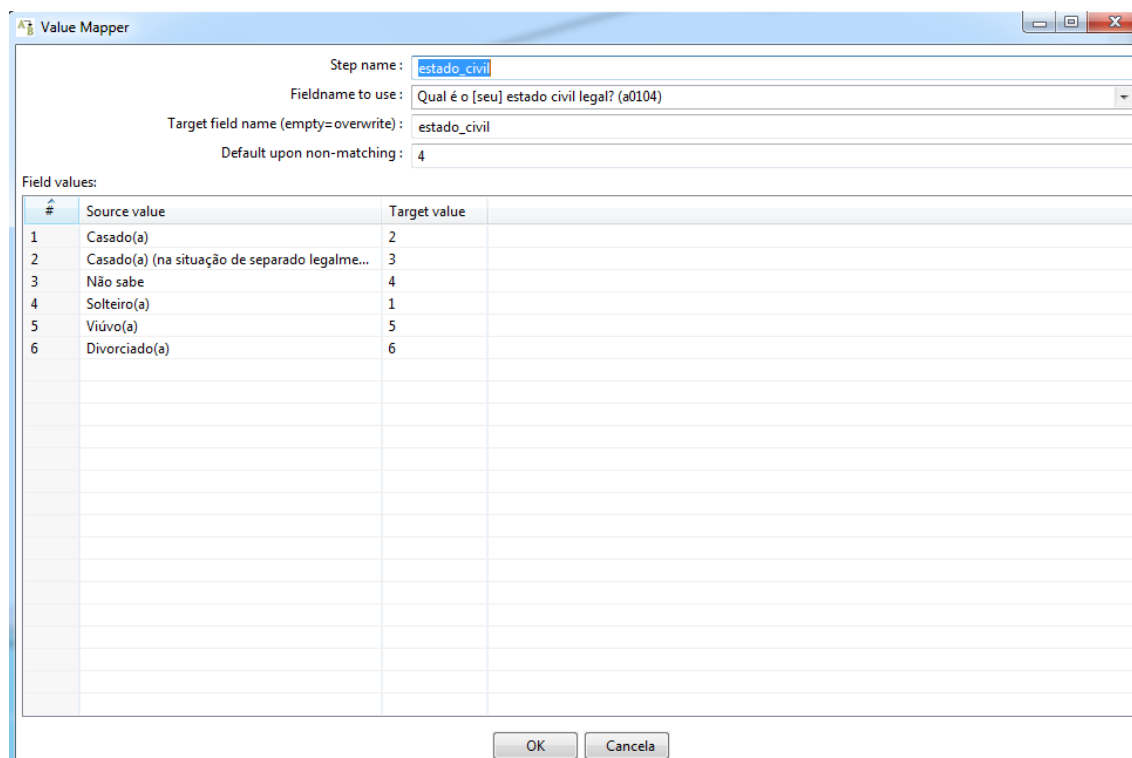


Figura 26- Mapeamento do Estado Civil.

Após o mapeamento de valores ter sido realizado um a um e antes de escrever os registos na tabela de factos, houve a necessidade de realizar um casting em alguns dos registos (Figura 27). Esta situação deveu-se aos valores serem lidos como strings no Excel mas escritos como inteiros na tabela de factos, apesar do *value mapper* realizar a correspondência das strings em números, estes não deixam de ser um carácter e não um inteiro. O GeoKettle contém uma ferramenta que faz o pretendido, usando o *Select/Rename values* temos a possibilidade de seleccionar e renomear campos ou apenas seleccionar e mudar o tipo de dado de alguns campos.

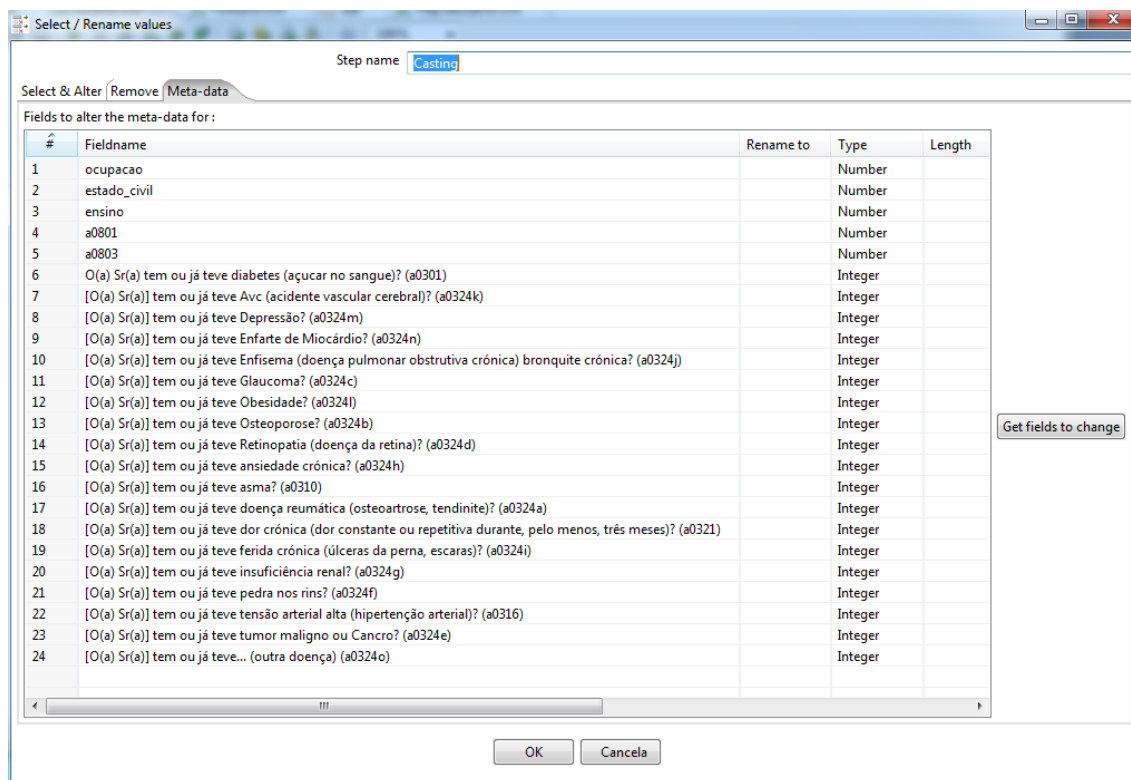


Figura 27 - Operações de casting.

A partir deste momento a cadeia de dados extraída e transformada está pronta para ser escrita na tabela de factos. Esta tarefa é realizada utilizando a ferramenta *Table Output* (Figura 28). A escrita é realizada usando uma ligação ODBC ao PostgreSQL, escolhendo-se o Schema e a tabela de destino será realizada uma eliminação de todos os registos da tabela de factos cada vez que este processo é repetido. A opção de *commit size* permite escolher quantos registos são escritos de cada vez na tabela. É uma opção que define o tamanho de um buffer onde vão sendo armazenados valores para irem sendo escritos, visto o inquérito ter apenas 1517 registos, optou-se por escrever tudo de uma só vez. O *Table Output* possui também a opção de utilizar um editor de SQL (Figura 29) para criar os campos a escrever na tabela de destino. Por último, a ferramenta tem a capacidade de consultar o tipo de campos existentes na tabela de destino e sugerir/realizar os devidos ajustes.

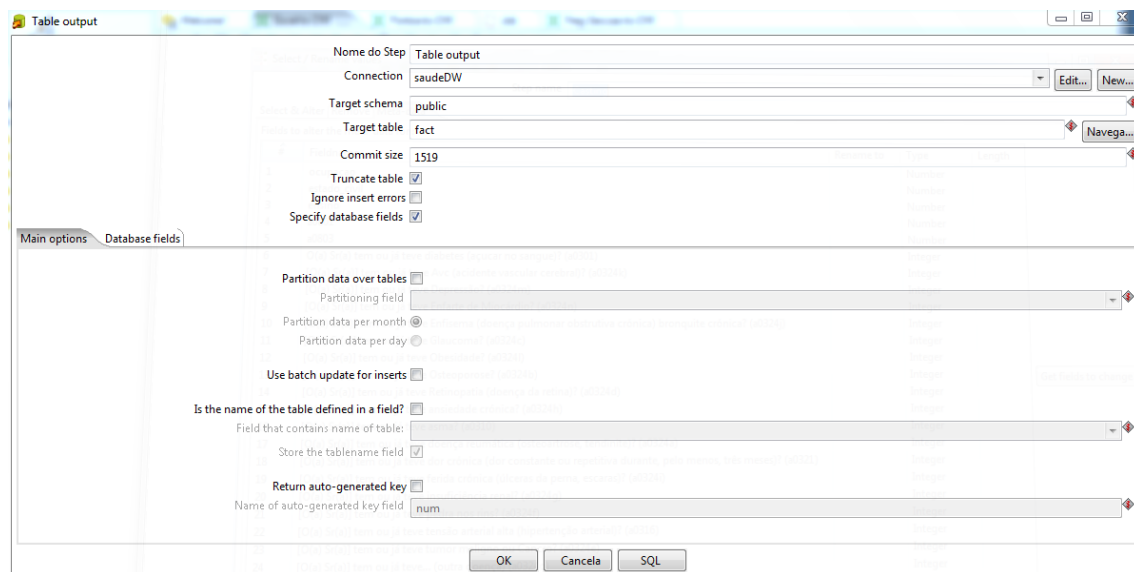


Figura 28 - Table Output.

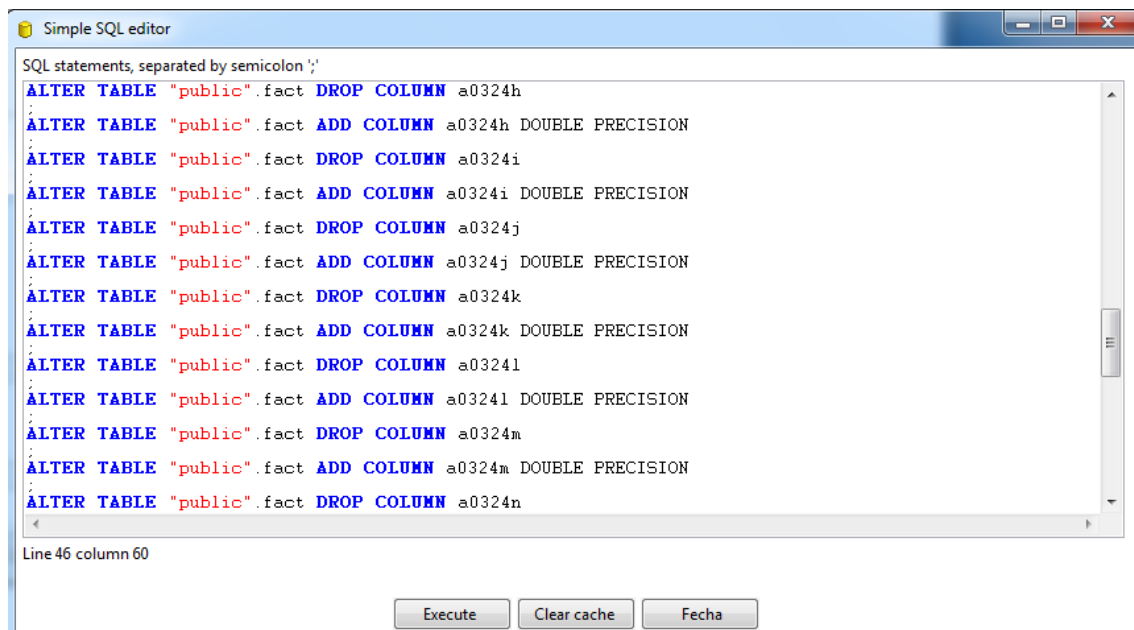


Figura 29 - Sql Editor.

De seguida realiza-se o mapeamento dos valores recebidos na *stream* dos passos anteriores da transformação, nos respetivos campos da tabela de factos (Figura 30).

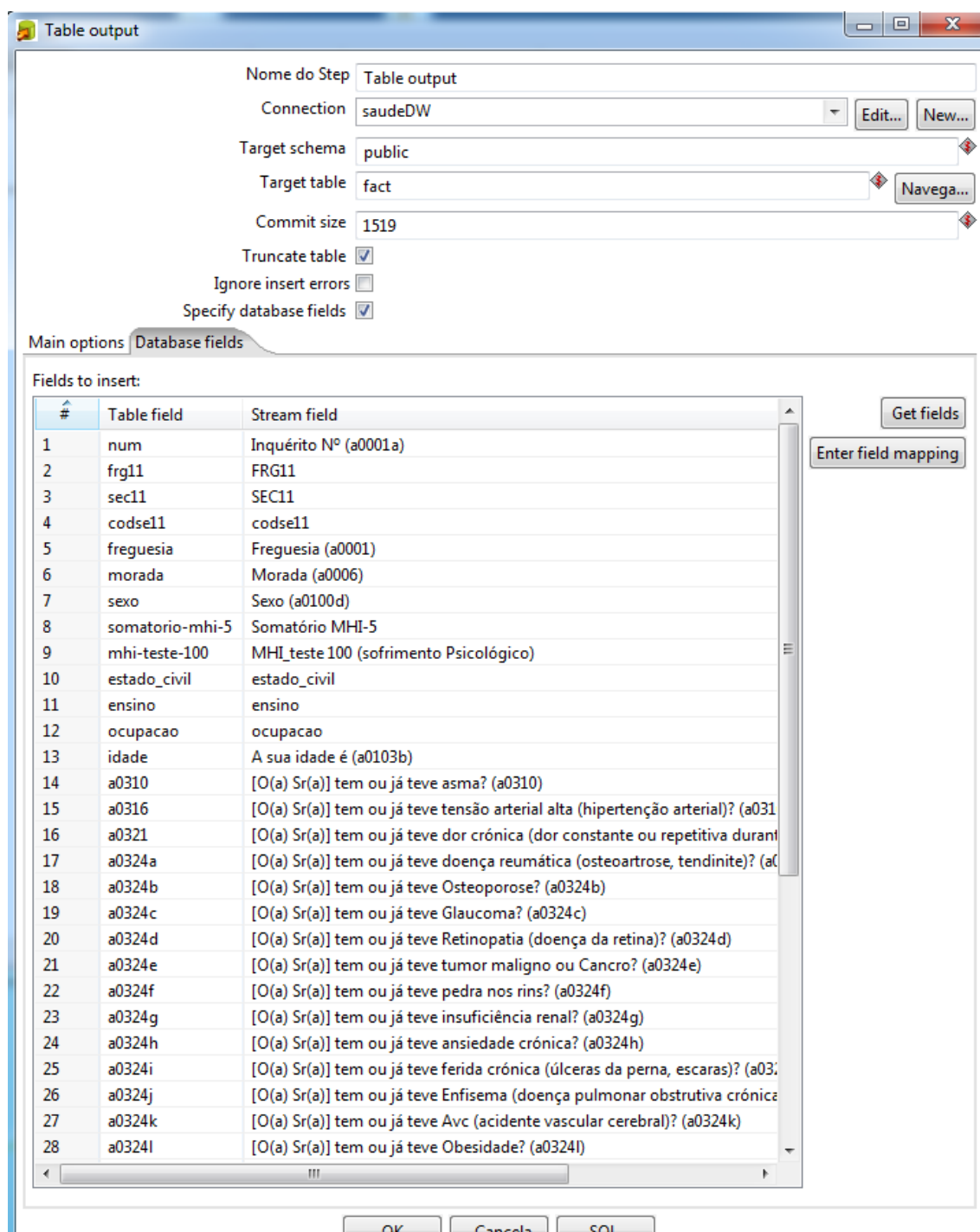


Figura 30 - Mapeamento de valores.

De todo o processo realizado, verificou-se (Figura 31) que ficou apenas um campo na tabela de factos por preencher. O campo “Ponto” (coordenadas geométricas onde foi realizada a entrevista) será preenchido posteriormente na

próxima transformação pois esta informação encontra-se numa shapefile (Figura 15).

A transformação Pontos-to-DW (Figura 32) é realizada em quatro passos. O primeiro passo vai efetuar a leitura das coordenadas de cada entrevista no shapefile (Figura 33). Além das coordenadas, cada ponto da shapefile está mapeado com o respetivo número de inquérito realizado sendo esta informação utilizada para fazer o update no registo correto da tabela de factos. O segundo passo configura o SRS, que é uma propriedade que define o sistema de referência espacial tendo-se neste caso optou-se por utilizar o valor devolvido da shapefile (Figura 34).

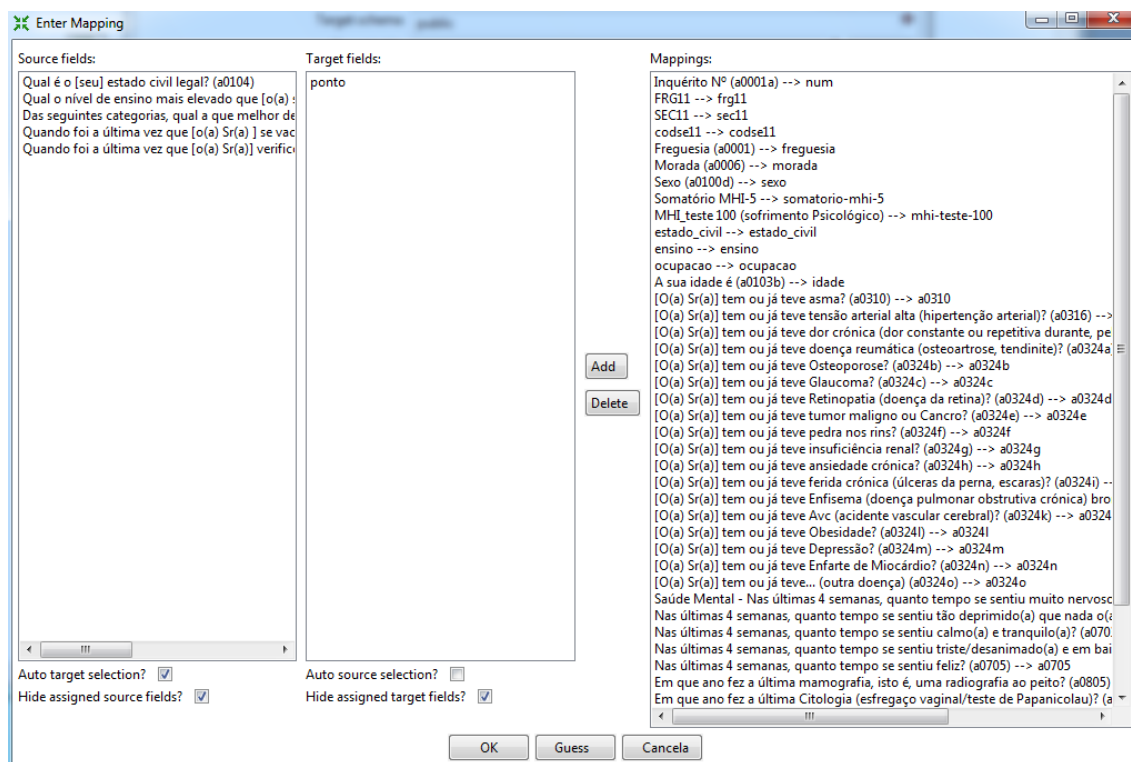


Figura 31 - Mapeamento manual.

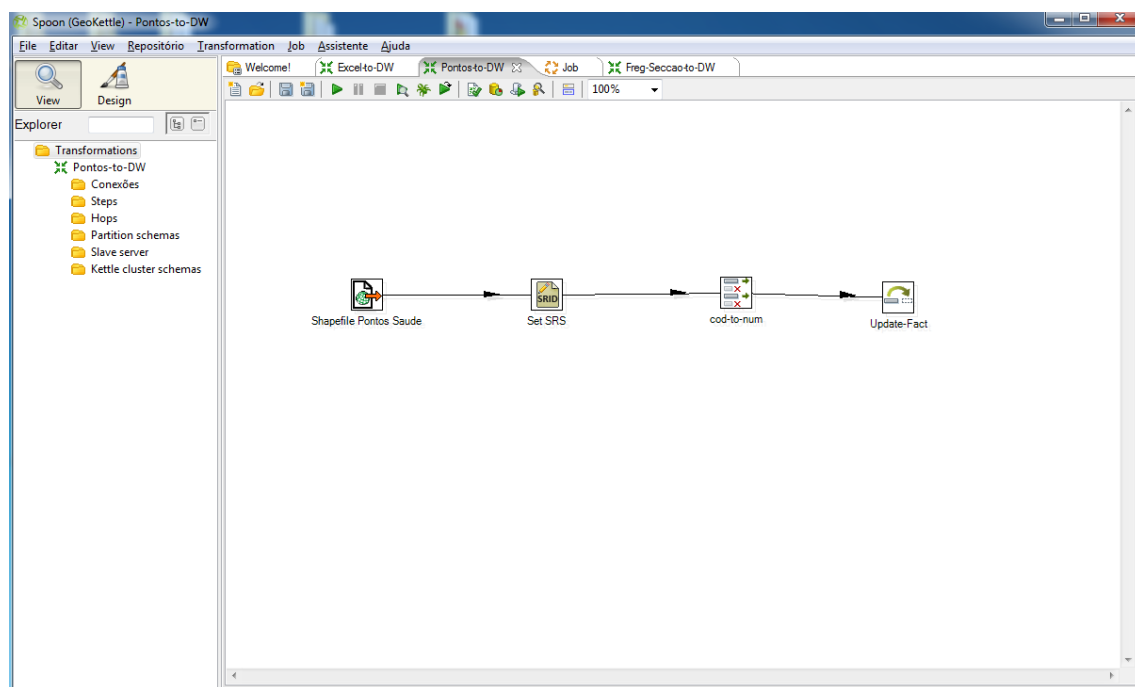


Figura 32 - Transformação Pontos-to-DW.

Nome do Step: Shapefile Pontos Saude

Nome do arquivo: C:\Users\silves\Desktop\Novos Dados\Inqueritos-geo\pontos

Filename is defined in a field? ☐

Pass through fields from previous step? ☐

Step to read filenames from:

Get filename from field:

Encoding:

Limit size: 0

Add rownr? (L...) ☐ Fieldname of rownr:

OK Preview Cancela

Examine preview data

Standard view Geographic view

Rows of step: Shapefile Pontos Saude (1000 rows)

#	the_geom	cod	X	Y
1	POINT (-8.8260399877142 41.700448595223)	17	-8,8	41,7
2	POINT (-8.8300329877082 41.6977059952235)	101	-8,8	41,7
3	POINT (-8.82255769877196 41.6973661952234)	248	-8,8	41,7
4	POINT (-8.8260159877134 41.6954254952237)	263	-8,8	41,7
5	POINT (-8.82327349877053 41.6953296952234)	282	-8,8	41,7
6	POINT (-8.82391419877168 41.6949810952235)	506	-8,8	41,7
7	POINT (-8.82391419877168 41.6949810952235)	558	-8,8	41,7
8	POINT (-8.8244929877165 41.7096469952226)	665	-8,8	41,7
9	POINT (-8.82746669877129 41.7015422952235)	756	-8,8	41,7
10	POINT (-8.82608439877141 41.7020674952232)	774	-8,8	41,7
11	POINT (-8.82788469877119 41.6921309952239)	779	-8,8	41,7
12	POINT (-8.82587009877139 41.6934098952239)	1026	-8,8	41,7
13	POINT (-8.82587009877139 41.6934098952239)	1027	-8,8	41,7
14	POINT (-8.82463709877164 41.7010678952234)	1028	-8,8	41,7
15	POINT (-8.82587009877139 41.6934098952239)	1029	-8,8	41,7
16	POINT (-8.82587009877139 41.6934098952239)	1031	-8,8	41,7
17	POINT (-8.82457159877161 41.7079575952227)	125	-8,8	41,7
18	POINT (-8.82494759877154 41.7068371952229)	126	-8,8	41,7
19	POINT (-8.82408469877168 41.6972806952233)	130	-8,8	41,7
20	POINT (-8.82753889877124 41.6957666952237)	131	-8,8	41,7
21	POINT (-8.82752139877124 41.7007949952233)	132	-8,8	41,7
22	POINT (-8.82449309877168 41.6970189952233)	134	-8,8	41,7
23	POINT (-8.82408469877168 41.6972806952233)	135	-8,8	41,7
24	POINT (-8.82408469877168 41.6972806952233)	137	-8,8	41,7
25	POINT (-8.82653279877131 41.6940938952237)	14	-8,8	41,7
26	POINT (-8.82408469877168 41.6972806952233)	140	-8,8	41,7
27	POINT (-8.82647579877136 41.6938911952237)	15	-8,8	41,7
28	POINT (-8.82255769877196 41.6973661952234)	153	-8,8	41,7

Fecha

Figura 33 - Leitura do shapefile.

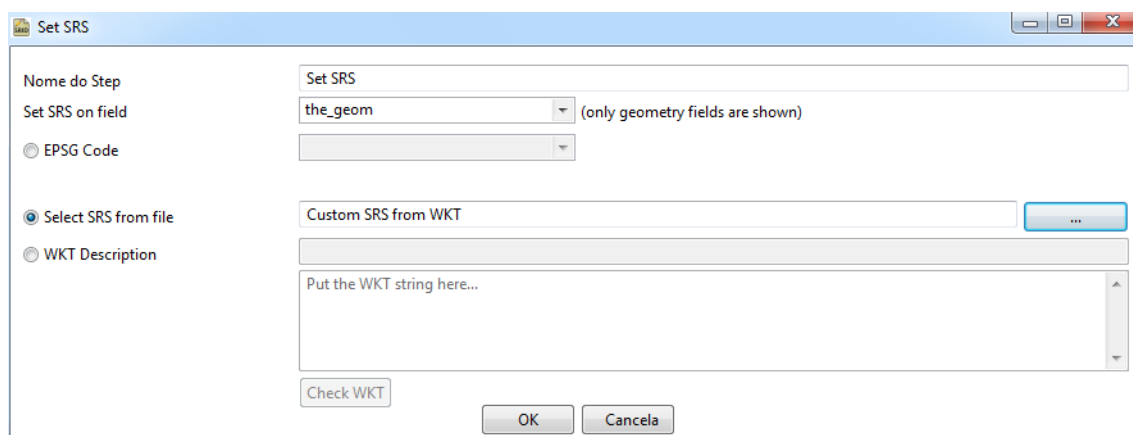


Figura 34 – Seleção do Sistema de Coordenadas.

Seguidamente usou-se a ferramenta *Select/Rename values* para renomear o nome do campo geométrico devolvido pela shapefile para o nome utilizado na tabela de factos (Figura 35).

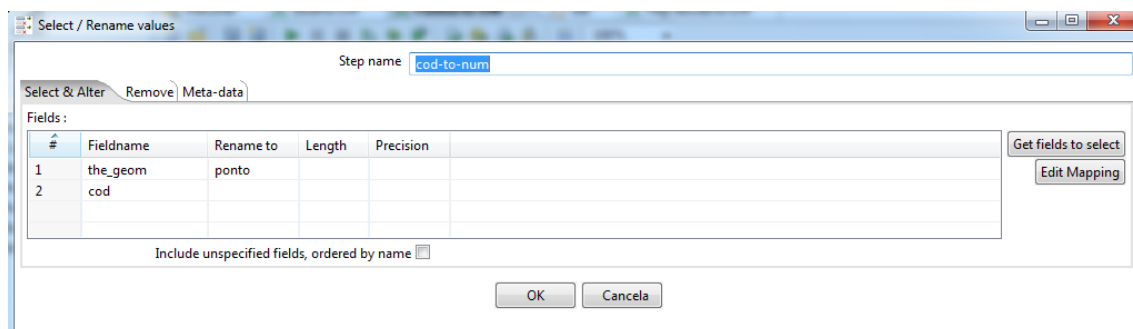


Figura 35 - Renomeação de campos.

No último passo realiza-se o update na tabela de factos onde será escrita a coordenada no campo “ponto” quando o número devolvido da shapefile é igual ao registo da tabela de factos (Figura 36). Apesar de, na visualização de dados e no processo OLAP este campo geométrico não ser utilizado, optou-se por realizar a sua importação para a data warehouse de forma a possibilitar num trabalho futuro a utilização do mesmo.

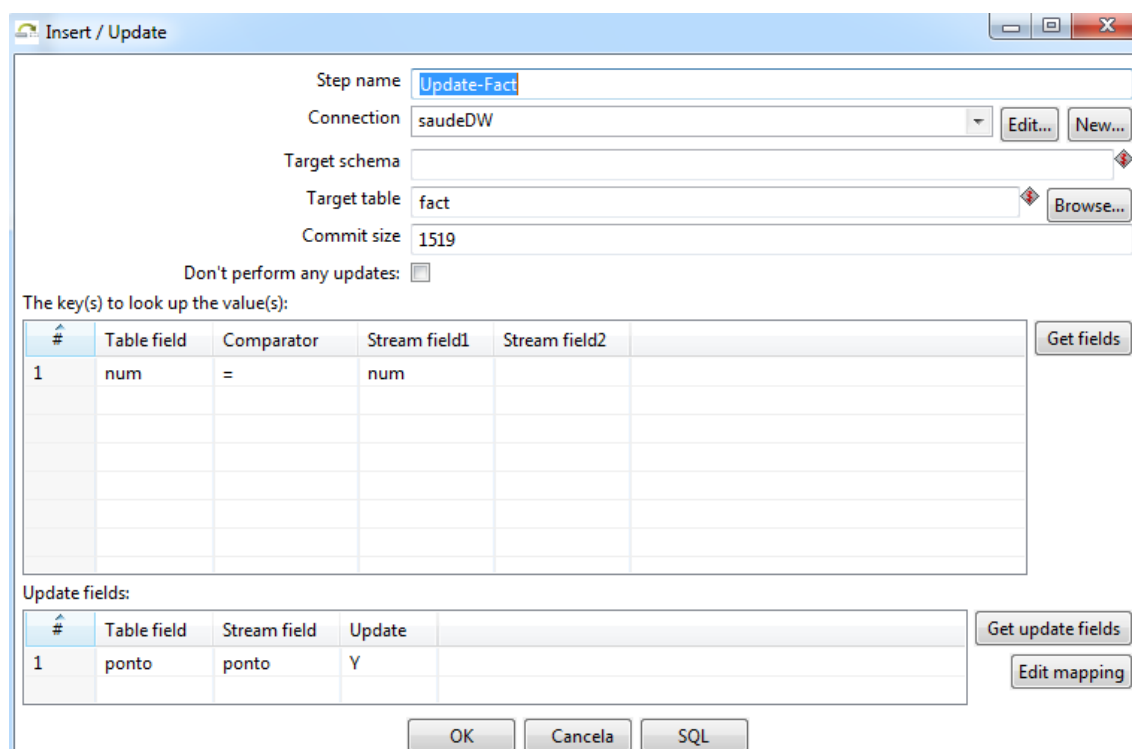


Figura 36 - Update na tabela de factos.

A última transformação do GeoKettle carrega os códigos identificativos e polígonos delimitadores das freguesias e secções das freguesias nas respetivas tabelas de dimensão da data warehouse (Figura 37).

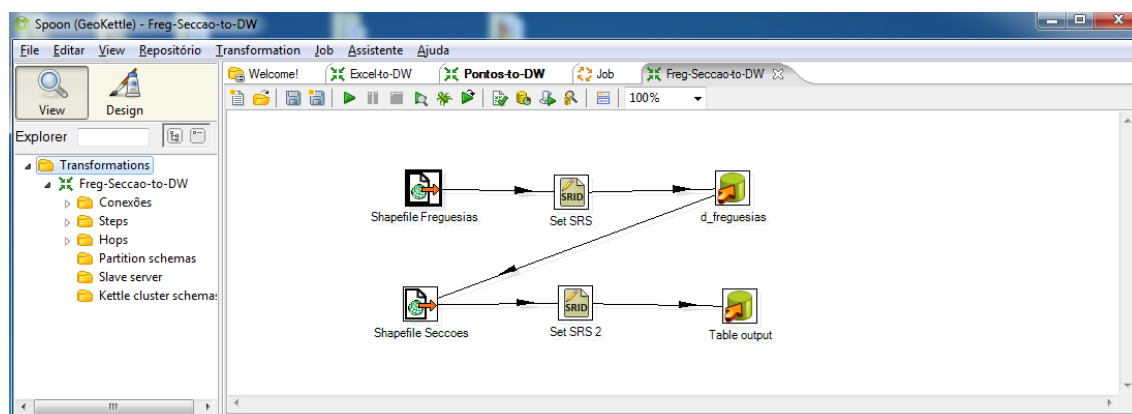


Figura 37 - Transformação Freg-Seccao-to-DW.

O processo de leitura das shapefiles é idêntico ao utilizado na transformação anterior. De forma a poder relacionar cada entrevista existente na tabela de factos com a freguesia e secção, além dos polígonos, foram igualmente importados da shapefile os campos numéricos identificativos de cada freguesia e secção. A figura 38 exemplifica de que forma é realizado este mapeamento para o caso das secções.

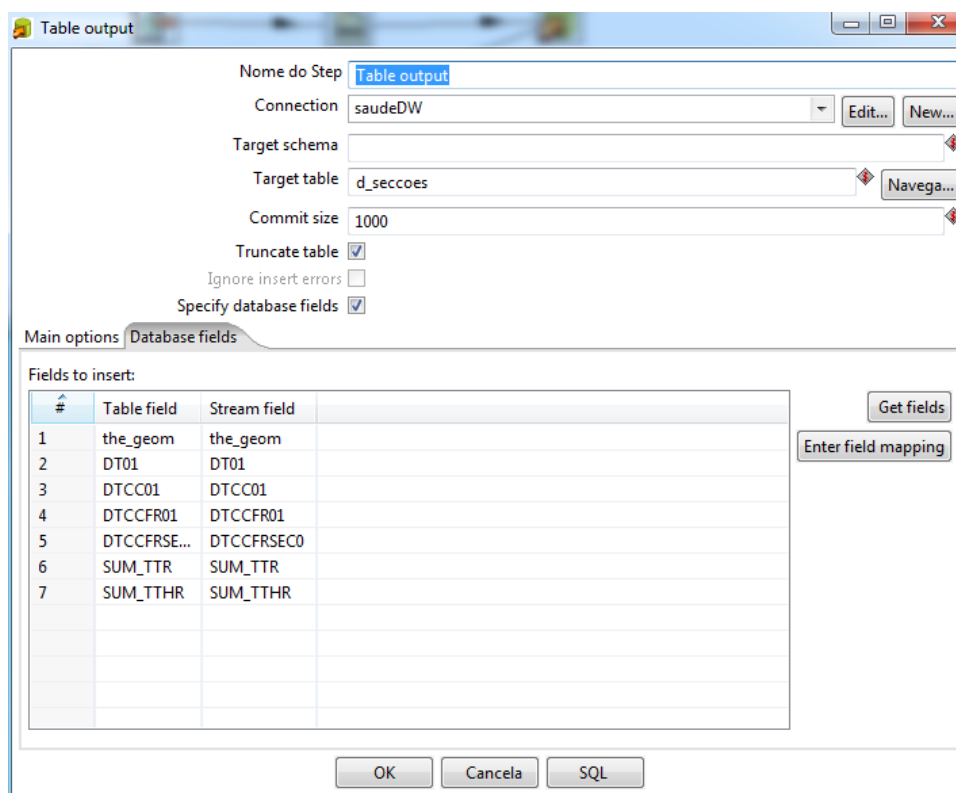


Figura 38 - Table output na tabela de dimensão d_seccoes.

Foi ainda necessário recorrer a um script SQL (Figura 39) para efetuar o mapeamento entre as chaves estrangeiras a0801 (vacinação contra gripe) e a0803 (medição de tensão arterial) da tabela de factos e as respetivas tabelas de dimensão d_vacina e d_exame_tensao.

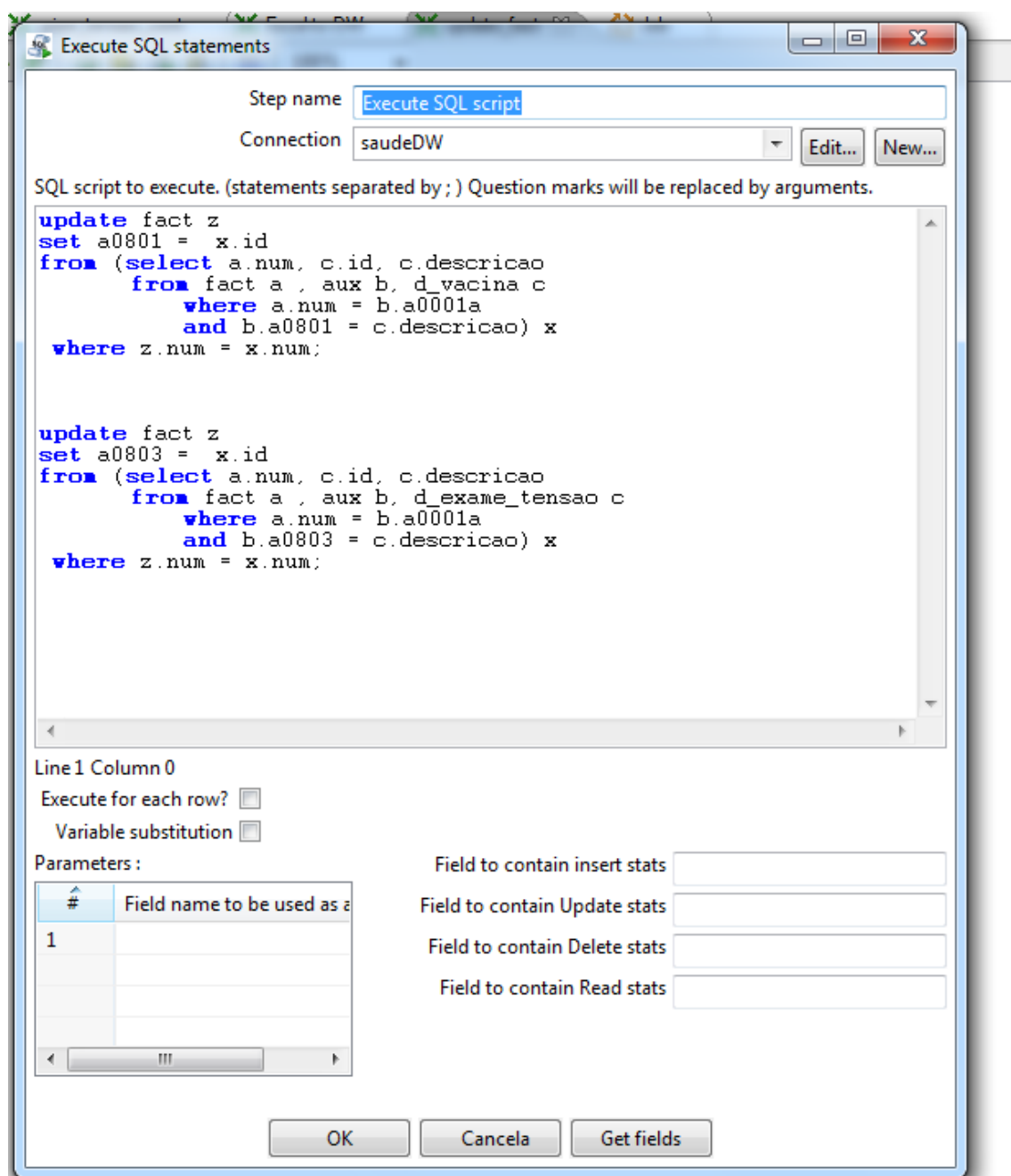


Figura 39 Atualização de tabela de factos

3.7 O Processo SOLAP

O processo SOLAP foi criado recorrendo ao motor GeoMondrian. Este motor permite combinar dados alfanuméricos com dados geográfico permitindo a navegação nos dados tendo uma perspetiva diferente. Neste estudo o objetivo passou por tentar visualizar a ocorrência geográfica de diversas doenças. Para satisfazer esta necessidade, o cubo criado teria que conter uma dimensão geográfica (união das freguesias e respetivas secções).

O GeoMondrian é disponibilizado num ficheiro do tipo *war* o qual é instalado no servidor Tomcat. O processo de instalação é bastante simples, bastando aceder à área de configuração do servidor TomCat e executar o deploy do ficheiro *geomondrian.war* (Figura 40). Os ficheiros *war* contêm componentes web baseados em java e aplicações passíveis de correr em servidores web, são formatados da mesma forma que os ficheiros *jar* mas incluem informações adicionais que informam o servidor web sobre que classes *java servlet* correr. Os ficheiros WAR podem conter classes *server-side* (tais como carrinhos de compra), recursos web estáticos (html, imagens, ficheiros de som) e classes *client-side* (applets).

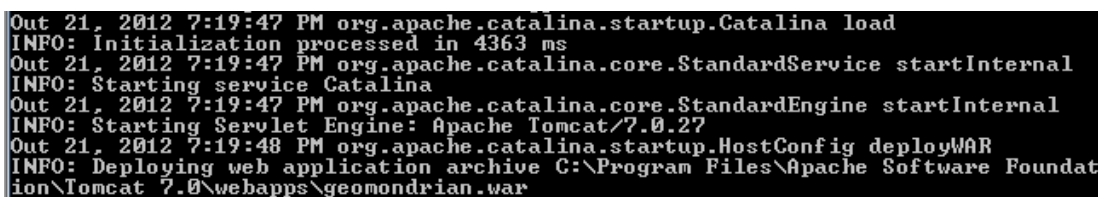
A screenshot of a Tomcat log window with a black background and white text. The logs show the startup sequence of the Catalina service, including initialization, starting the service, and deploying the web application archive C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\geomondrian.war. The text is as follows:
Out 21, 2012 7:19:47 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 4363 ms
Out 21, 2012 7:19:47 PM org.apache.catalina.core.StandardService startInternal
INFO: Starting service Catalina
Out 21, 2012 7:19:47 PM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/7.0.27
Out 21, 2012 7:19:48 PM org.apache.catalina.startup.HostConfig deployWAR
INFO: Deploying web application archive C:\Program Files\Apache Software Foundat
ion\Tomcat 7.0\webapps\geomondrian.war

Figura 40 - Tomcat fazendo o deploy do *geomondrian.war*.

O motor SOLAP Geomondrian vem acompanhado do software Workbench (Figura 41) que é uma aplicação desktop que facilita a construção dos cubos. Esta aplicação consulta as tabelas da datawarehouse usando uma

ligação ODBC, permitindo fazer a criação e customização dos cubos. A abordagem inicial na criação dos cubos passou por utilizar esta ferramenta, mas verificou-se que à medida que o cubo ia crescendo, a ferramenta ia-se tornando muito lenta chegando ao ponto em que deixou de ser rentável a sua utilização. Apesar deste facto, a utilização do Workbench teve um aspeto positivo pois a ferramenta cria o ficheiro em formato XML que é lido pelo GeoMondrian. Sendo assim bastou criar a dimensão Geo e mais algumas dimensões com dados alfanuméricos com a ferramenta para perceber qual a lógica e sintaxe de construção do ficheiro XML, continuando depois a construção deste ficheiro recorrendo a um simples editor de texto (notepad++) para finalizar e corrigir alguns aspetos do cubo.

A ferramenta Workbench é mais uma que derivou da sua homóloga na suite Pentaho, mas fica-se com a ideia que serve apenas para criar os cubos e pouco mais, pois não permite a utilização da funcionalidade de publicação do cubo no Tomcat (ao contrário da versão para Pentaho) nem permite a utilização da tool *MDX Querys* que permitiria executar consultas MDX a fim de testar o bom funcionamento do cubo (esta funcionalidade funciona bem no Workbench da suite Pentaho). O ficheiro XML pode conter um ou mais cubos, estando todos hierarquicamente dentro do mesmo *schema*. A construção do cubo inicia-se escolhendo a tabela de factos com que o cubo irá trabalhar. O passo seguinte passa pela criação das diversas dimensões e finalmente as medidas. Foi decidido a criação de três cubos para análise dos dados do inquérito: cubo de análise de doenças crónicas, doenças mentais e cubo de análise de bem-estar. Esta divisão permite que a análise de dados relativos às doenças seja realizada por categoria de doença.

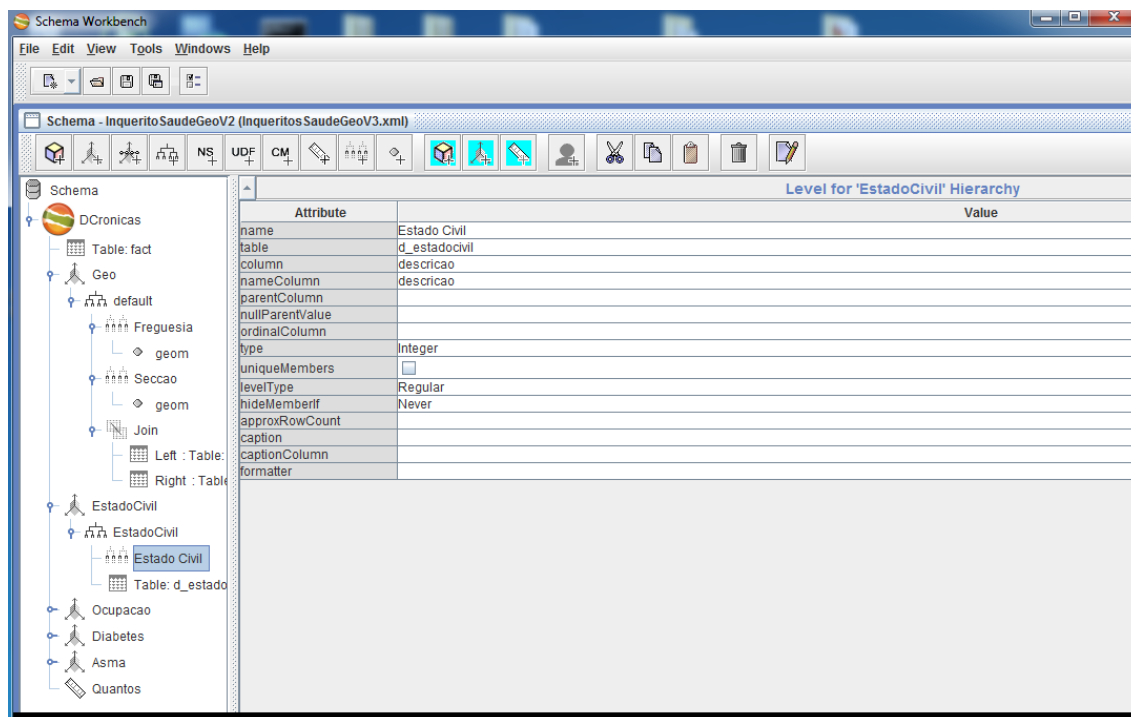


Figura 41 - Schema Workbench.

Na figura 42 apresenta-se um excerto do cubo de análise de doenças crónicas. O esquema chama-se “InqueritoSaudeGeoV2” e contém um cubo chamado “dcronicas” que usa a tabela “fact” (tabela de factos). No exemplo podemos analisar quatro dimensões, a primeira dimensão é Geo, que permite que os dados do cubo sejam visualizados usando uma perspetiva geográfica; a dimensão é composta de uma agregação de duas dimensões (secções e freguesias) esta agregação permite que seja realizado um drill-down ou roll-up geométrico em que inicialmente os dados são visualizados ao nível da freguesia, mas graças à agregação entre as duas tabelas a dimensão geo permite que seja realizado um drill de forma a decompor cada freguesia nas suas respetivas secções. Isto é possível devido à operação de join entre as tabelas *d_seccoes* e *d_freguesias*. O relacionamento com a tabela de factos é realizado através do campo *codse11*. O campo *codse11* da tabela de factos é chave estrangeira do campo *dtccfrsec0* da tabela *d_seccoes*. As dimensões seguintes permitem a visualização dos dados com uma perspetiva não geográfica, por exemplo, analisar a medida “quantos diabéticos existem numa freguesia ou secção de freguesia por estado civil e ocupação”.

```
<Schema name="InqueritoSaudeGeoV2">
  <Cube name="DCronicas" cache="true" enabled="true">
    <Table name="fact">
    </Table>
    <Dimension foreignKey="codse11" name="Geo">
      <Hierarchy hasAll="true" primaryKey="dtccfrsec0" primaryKeyTable="d_seccoes">
        <Join leftKey="dtccfr01" rightKey="dtccfr01">
          <Table name="d_seccoes">
          </Table>
          <Table name="d_freguesias" schema="public">
          </Table>
        </Join>
        <Level name="Freguesia" table="d_freguesias" column="ff_dsg_res" type="String"
uniqueMembers="true" levelType="Regular" hideMemberIf="Never">
          <Property name="geom" column="the_geom" type="Geometry">
          </Property>
        </Level>
        <Level name="Seccao" table="d_seccoes" column="dtccfrsec0" type="String"
uniqueMembers="true" levelType="Regular" hideMemberIf="Never">
          <Property name="geom" column="the_geom" type="Geometry">
          </Property>
        </Level>
      </Hierarchy>
    </Dimension>

    <Dimension type="StandardDimension" foreignKey="estado_civil" name="EstadoCivil" caption="Estado
Civil">
      <Hierarchy name="EstadoCivil" hasAll="true" primaryKey="id" caption="Estado Civil">
        <Table name="d_estadocivil" schema="public">
        </Table>
        <Level name="Estado Civil" table="d_estadocivil" column="descricao" nameColumn="descricao"
type="Integer" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>

    <Dimension type="StandardDimension" foreignKey="ocupacao" name="Ocupacao"
caption="Ocupa&#231;&#227;o">
      <Hierarchy name="Ocupacao" hasAll="true" allMemberName="AllOcupacao" primaryKey="id"
caption="Ocupacao">
        <Table name="d_ocupacao" schema="public">
        </Table>
        <Level name="Ocupacao" table="d_ocupacao" column="descricao" nameColumn="descricao"
type="Integer" uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>

    <Dimension type="StandardDimension" foreignKey="diabetes" name="Diabetes" caption="Diabetes">
      <Hierarchy name="Diabetes" hasAll="true" allMemberName="AllDiabetes" primaryKey="id"
caption="Diabetes">
        <Table name="MR1" schema="public">
        </Table>
        <Level name="Diabetes" table="MR1" column="descricao" nameColumn="descricao" type="Integer"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>
  </Cube>
</Schema>
```

```
</Dimension>

<Measure name="Quantos" column="num" aggregator="count" caption="Quantos" visible="true">
</Measure>
</cube>
</schema>
```

Figura 42 - Excerto XML do Cubo

3.8 Business Intelligence – Visualização de Dados Analíticos

A visualização dos dados analíticos é possível graças à biblioteca JPivot. É esta biblioteca que realiza a renderização da tabela SOLAP e permite a navegação no cubo aos utilizadores. O JPivot utiliza o GeoMondrian como servidor SOLAP tendo acesso ao cubo através de XMLA. A linguagem de consultas utilizada é o MDX. O MDX está para o OLAP da mesma forma que o SQL está para as bases de dados relacionais. Graças ao JPivot é possível executar consultas MDX a partir de uma página estática. O JPivot está entre o servidor web (Tomcat) e o servidor SOLAP funcionando como middleware. O JPivot recorre a um container servlet que gera dinamicamente o HTML obtido através da execução de código onde as consultas MDX e suas respostas são enviadas/recebidas. Podemos dizer então que o JPivot é uma ferramenta de análise de dados multidimensionais. Foi obtida e instalada no Tomcat uma cópia do projeto GeoMondrian. Esta pasta foi instalada no diretório webapps do tomcat (Figura 43) estando assim acessível através do endereço <http://localhost:8080/geomondrianv3/>

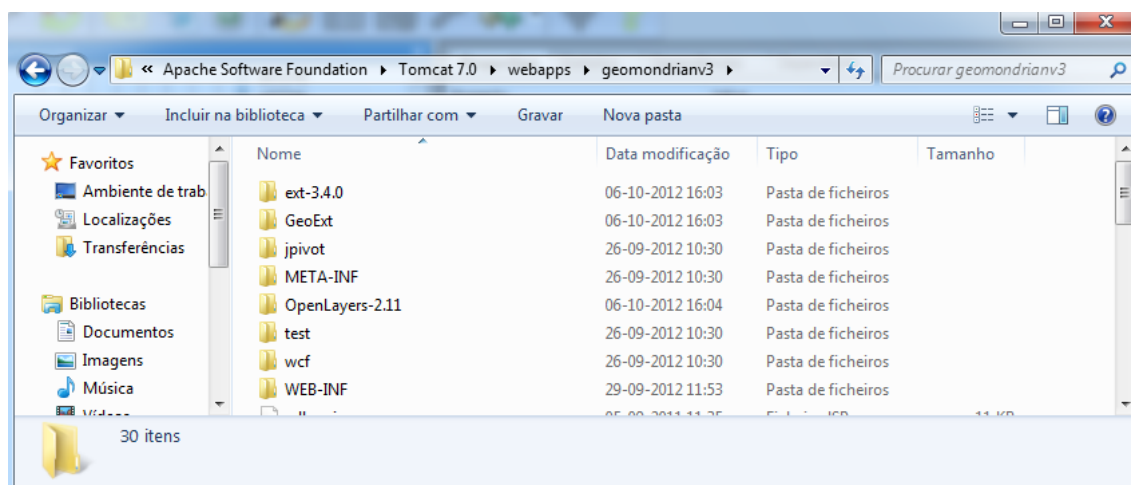


Figura 43 - Raiz do website GeoMondrian.

O GeoMondrian vem com alguns exemplos preparados e prontos a utilizar sobre um data warehouse exemplo (geofoodmart). Estes exemplos foram aproveitados e adaptados ao nosso caso de estudo. Houve necessidade de adaptar o exemplo da geofoodmart devido à falta de informação sobre como preparar um projeto de raiz. A documentação do GeoMondrian revelou-se pouco útil e foi necessário perceber como estaria estruturado o exemplo disponibilizado no GeoMondrian. O primeiro passo passou pelo estudo da estrutura de ficheiros e de como seria processada e visualizada a navegação nos cubos. Após este estudo, chegou-se à conclusão que a base de acesso aos cubos está no ficheiro “testpage.jsp” que recebe como parâmetro a consulta MDX a ser processada “http://localhost:8080/geomondrianv3/testpage.jsp?query=geomondrian_spatial cube2”.

O ficheiro testpage.jsp contém um formulário que invoca e processa a consulta mdx passada em parâmetro no motor SOLAP. Os resultados da invocação são passados ao JPivot que os processa e cria dinamicamente uma tabela permitindo então ao utilizador navegar através do cubo. O parâmetro passado por argumento ao testpage.jsp é um ficheiro jsp que contém a consulta MDX que serve de base inicial (Figura 44).

```
<%@ page session="true" contentType="text/html; charset=ISO-8859-1" %>
<%@ taglib uri="http://www.tonbeller.com/jpivot" prefix="jp" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>

<jp:mondrianQuery id="query01" jdbcDriver="org.postgis.DriverWrapper"
jdbcUrl="jdbc:postgresql_postGIS://localhost/saudeDW?user=postgres&password=mtgsi" catalogUri="/WEB-INF/queries/InqueritosSaudeGeoV3.xml">
select {[Measures].[Quantos]} ON COLUMNS,
    Crossjoin(Hierarchize(Union(Union({[Geo].[All Geos]}}, [Geo].[All Geos].Children), [Geo].[All Geos].[Freguesia1].Children)), {[Diabetes].[AllDiabetes].[Sim]}) ON ROWS
from [DCronicas]

</jp:mondrianQuery>

<c:set var="title01" scope="session">Test Query uses GeoMondrian SOLAP server</c:set>
```

Figura 44 - Consulta MDX

Pode-se verificar que é neste ficheiro que está programada a MondrianQuery com a ligação à base de dados PostgreSQL através da utilização do driver java (ex: jdbc). É informado também qual o ficheiro XML que contém as definições do Cubo (catalogUri="/WEB-INF/queries/InqueritosSaudeGeoV3.xml") e sua localização, bem como a consulta MDX inicial a ser usada. Após o processamento desta consulta MDX inicial, a mesma pode ser alterada recorrendo para isso, ou ao editor MDX, ou ao Open OLAP Navigator ambos embutido no JPivot. Na figura 45 apresenta-se um exemplo onde foi efetuada uma consulta recorrendo a quatro dimensões.

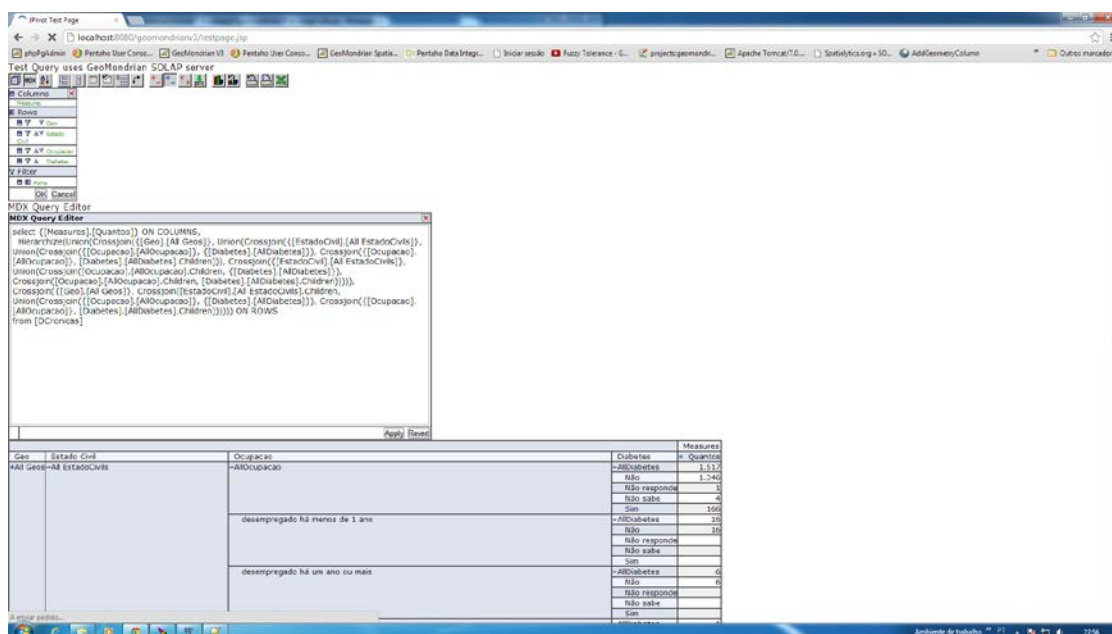


Figura 45 - Consulta efetuada no JPivot.

Através do Open OLAP Navigator, o utilizador pode customizar quais as dimensões a usar e aplicar filtros às mesmas (Figura 46). A utilização desta ferramenta é recomendável pois evita a criação manual de consultas MDX.

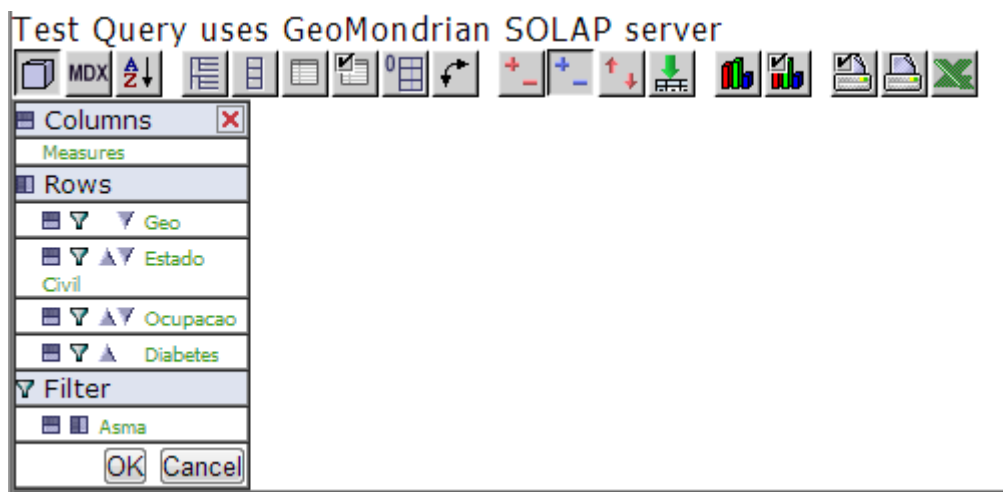


Figura 46 - Open OLAP Navigator.

A título de exemplo apresenta-se na figura 47 a consulta MDX criada no OLAP Navigator, da consulta da figura 45 em que se analisa quantos responderam “sim” à doença “asma” tendo em conta as medidas Geo, Estado Civil, Ocupação e Diabetes.

```
select {[Measures].[Quantos]} ON COLUMNS,
  Hierarchize(Union(Crossjoin({[Geo].[All Geos]}, Union(Crossjoin({[EstadoCivil].[All
EstadoCivils]}, Union(Crossjoin({[Ocupacao].[AllOcupacao]}, {[Diabetes].[AllDiabetes]}),
Crossjoin({[Ocupacao].[AllOcupacao]}, [Diabetes].[AllDiabetes].Children))),
Crossjoin({[EstadoCivil].[All EstadoCivils]}, Union(Crossjoin([Ocupacao].[AllOcupacao].Children,
{[Diabetes].[AllDiabetes]}), Crossjoin([Ocupacao].[AllOcupacao].Children,
[Diabetes].[AllDiabetes].Children))))), Crossjoin({[Geo].[All Geos]}, Crossjoin([EstadoCivil].[All
EstadoCivils].Children, Union(Crossjoin({[Ocupacao].[AllOcupacao]}, {[Diabetes].[AllDiabetes]}),
Crossjoin({[Ocupacao].[AllOcupacao]}, [Diabetes].[AllDiabetes].Children)))))) ON ROWS
from [DCronicas]
```

Figura 47 - Consulta MDX devolvida pelo OLAP Navigator

Depois de verificar o funcionamento do JPivot, o próximo passo passou por efetuar algumas alterações ao ficheiro testpage.jsp de forma a poder visualizar num mapa a dimensão Geo. Ao pressionar o botão “Show Properties” é acrescentada (à tabela dinâmica com os resultados), uma coluna “geom” que apresenta o polígono ou multi-polígono referente à dimensão geográfica.

Na figura 48 podemos verificar o resultado de uma consulta MDX onde se consulta quantos inquéritos responderam “sim” na doença “Diabetes” na

“Business Intelligence” sobre informação espacial: um caso de estudo

freguesia 1, onde a coluna apresenta um multi-polígono que resulta da agregação dos polígonos de todas as secções da freguesia 1.



Figura 48 - Visualização de multi-polígono da dimensão Geo.

Um dos objetivos iniciais deste projeto passava pela utilização do SOLAPLayers para a visualização em mapa das dimensões geométricas, mas verificou-se que o projeto SOLAPLayers foi descontinuado devido a estar disponível a sua utilização quando o projeto já tinha sido iniciado. Este facto veio alterar metodologia que estava inicialmente pensada. De forma a ultrapassar este contratempo, optou-se pela construção de um protótipo utilizando a API Javascript do OpenLayers. A API foi programada para utilizar um array de dados multidimensional onde cada linha contém dois valores: a medida quantos e o polígono. Este array é utilizado com a API OpenLayers de forma a desenhar no mapa os polígonos de acordo com uma legenda pré-definida.

Para permitir a visualização gráfica dos resultados, foram realizadas algumas alterações (anexo 6.1.1) ao ficheiro testpage.jsp de forma a embutir o mapa de Portugal. De forma a alimentar o array de dados com os polígonos ou multi-polígonos foi programado um ciclo for que percorre as linhas da tabela resultante da consulta MDX produzida dinamicamente pelo JPivot a fim de copiar os dados referentes às quantidades e aos polígonos (Figura 49).

```
<script language = "JavaScript">

    var linha;

    for (linha=3; linha<document.getElementsByTagName("table")[1].rows.length; linha++)
    {
        dados.push({ 'value':
document.getElementsByTagName("table")[1].rows[linha].cells[3].childNodes[0].data,
                    'geometry':
document.getElementsByTagName("table")[1].rows[linha].cells[1].childNodes[1].data
                    });

        //parseFloat(document.getElementsByTagName("table")[1].rows[linha].cells[3].childNodes[0].da
ta.replace(',','')),
        //html/body/form/p[2]/table/tbody/tr[3]/th[2]

    }
</script>
```

Figura 49 - Extração de Polígonos e Quantidades

A representação geográfica da Figura 50, reflete o resultado de uma consulta onde se analisa quantas respostas “sim” à doença “diabetes” foram respondidas nas freguesias 1, 2, 3 e 4. Consoante o número de respostas “sim” os polígonos são pintados de cor diferente. A vermelho representam-se polígonos onde a consulta obteve um resultado inferior a 10, a verde onde o resultado está situado entre 11 e 50 e a azul onde o resultado é superior a 51. Neste caso apesar de geograficamente a freguesia 2 ser mais reduzida, percebe-se que a consulta devolveu mais de 51 respostas positivas na freguesia 2, ao contrário das freguesias 1 e 4 onde foram obtidas entre 11 a 50 respostas positivas.

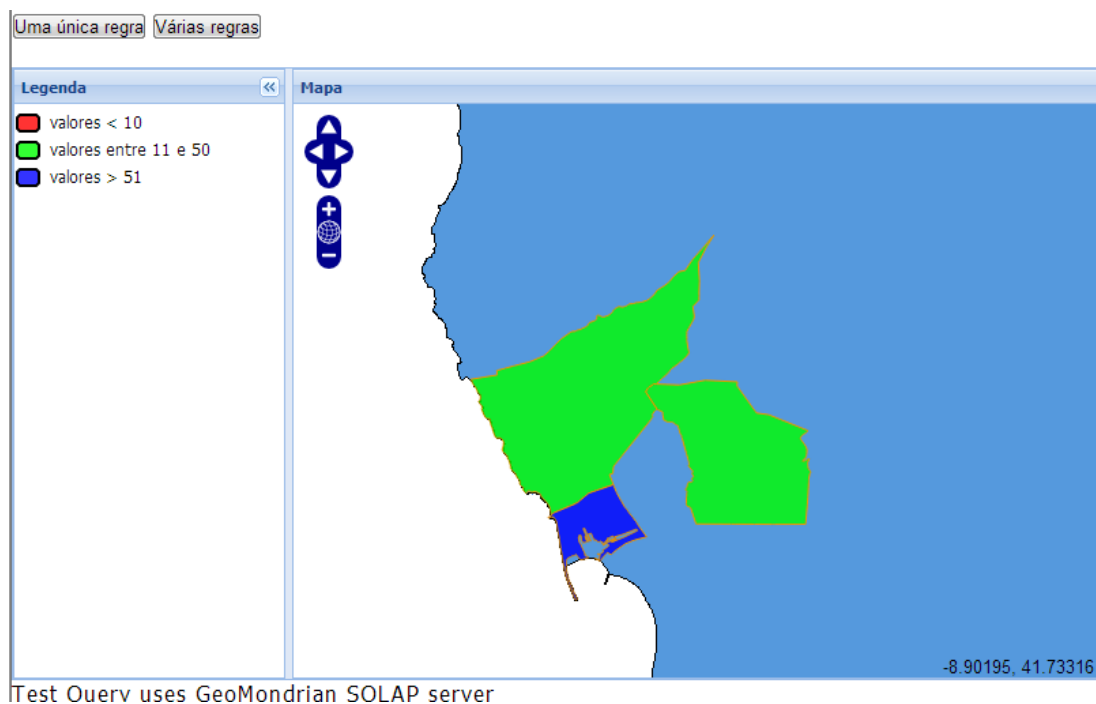


Figura 50 - Representação Geográfica da Informação

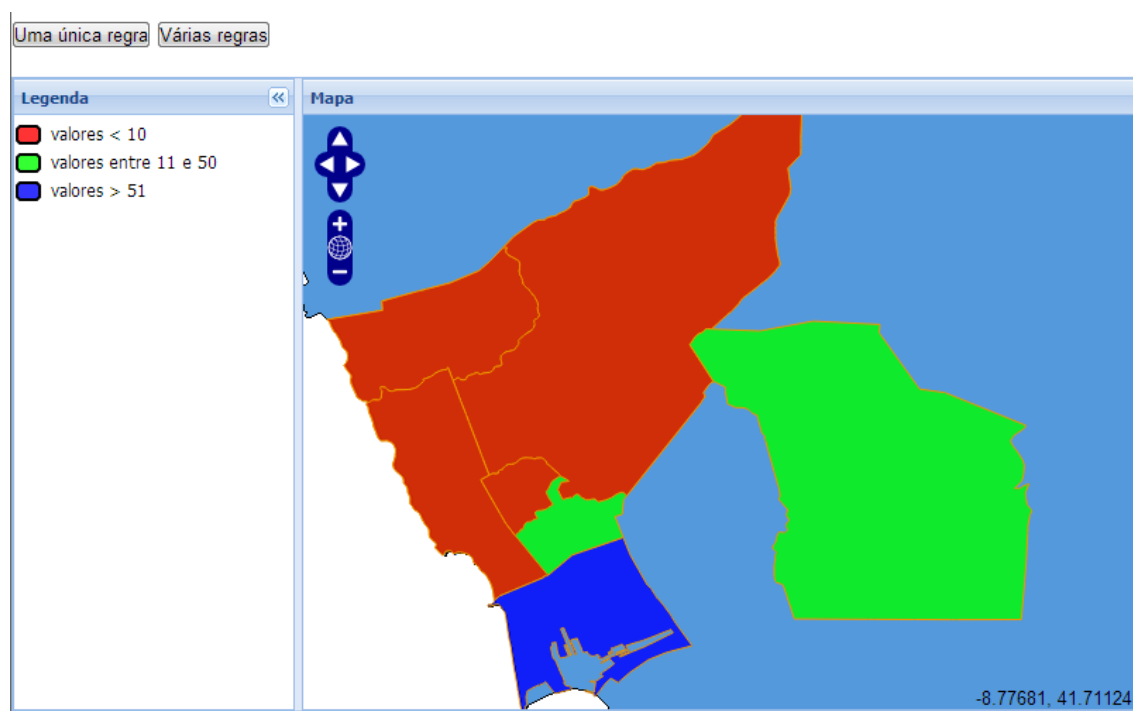


Figura 51 – Efeito Drill-Down.

Após a aplicação da operação Drill-Down apenas na freguesia 1, obtém-se uma representação geográfica com a divisão desta freguesia nas suas

secções. A operação Drill-Down permitiu verificar que em quatro secções foram obtidos valores inferiores a 10 e apenas em uma secção foram obtidos valores entre 11 e 50.

A figura 52 representa o resultado da aplicação de um Roll-Up na freguesia 1 e um Drill-Down na freguesia 4.

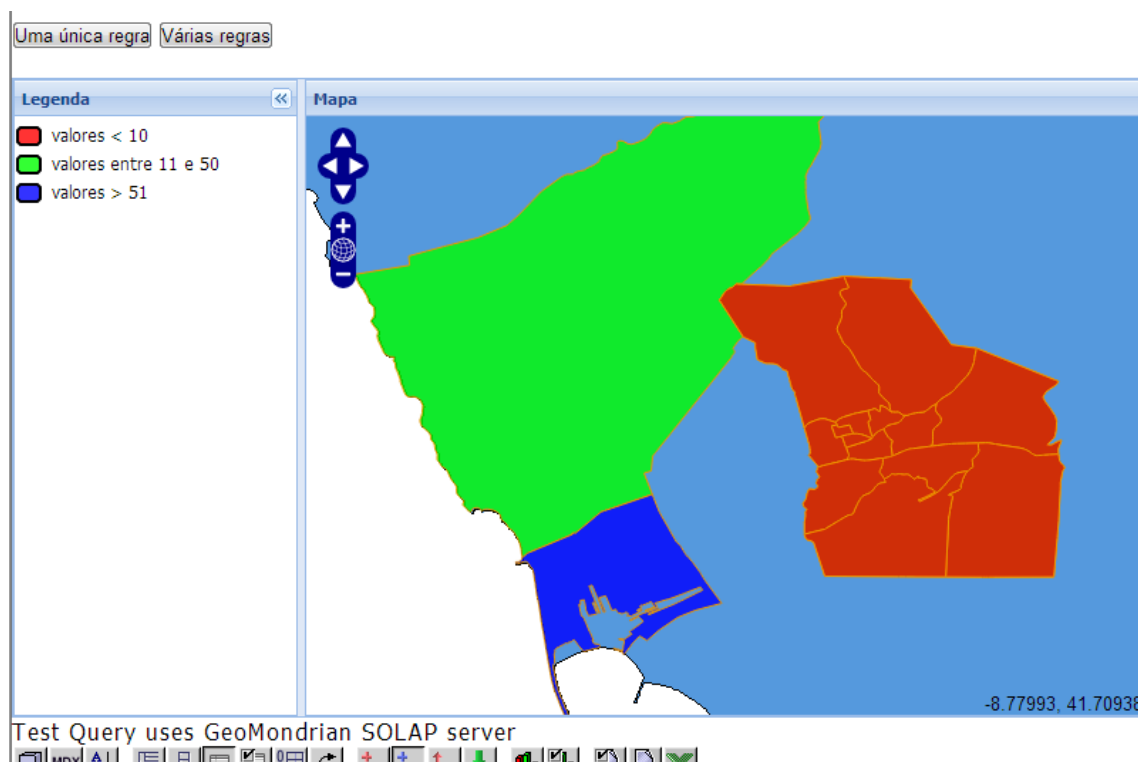


Figura 52 - Efeito Roll-Up e Drill-Down

A figura 53 representa o efeito Drill-Down aplicado a todas as freguesias. Todas as secções estão representadas geograficamente, como resposta à pergunta sobre “diabetes” visualiza-se geograficamente a azul que uma secção da freguesia 2 obtém valores superiores a 51, duas secções (uma da freguesia 1 e outra da freguesia 4) a verde obtém valores compreendidos entre 11 e 50, tendo as restantes secções obtido valores inferiores a 10.

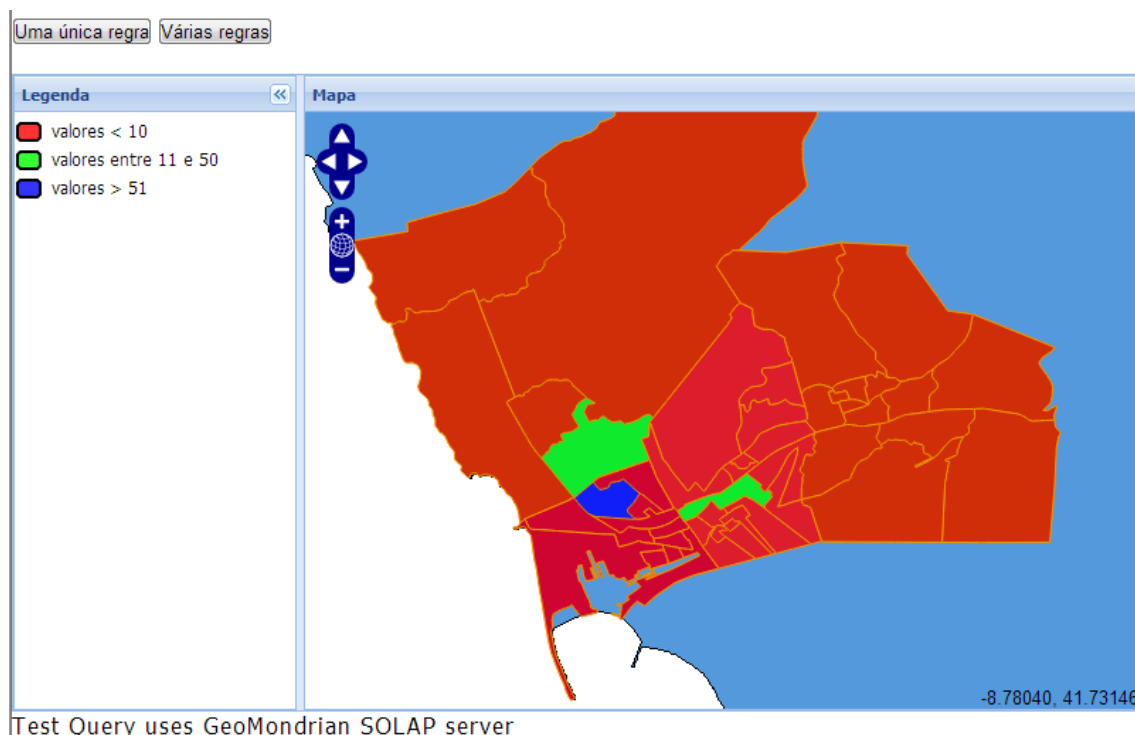


Figura 53 – Efeito Drill-Down em todas as Freguesias

3.9 Análise e Discussão

Após o estudo e construção do modelo multidimensional tendo como base os inquéritos providenciados pela Escola Superior de Saúde do Instituto Politécnico de Viana do Castelo em articulação com o Observatório de Saúde do Gabinete de Cidade Saudável, foi definida a estratégia para o processo de ETL. Os dados obtidos através dos inquéritos, foram extraídos, tratados e carregados na data warehouse. O processo de ETL permitiu efetuar a seleção das colunas consideradas relevantes, a tradução/codificação dos valores alfanuméricos e a junção de diversas fontes de dados (alfanuméricas e georreferenciadas). Após a experimentação do software GeoKettle no seu global realizando vários testes, conseguiu-se chegar a uma configuração que se revelou estável e fiável que foi de encontro aos objetivos propostos inicialmente. A utilização da mesma, não se revelou muito complicada, a quantidade de informação e exemplos encontrados na internet foi suficiente

para o processo de aprendizagem. Conclui-se então que a opção pelo GeoKettle para o processo de ETL foi acertada.

A construção dos cubos para o processo SOLAP foi efetuada com sucesso. Mas esta fase do projeto foi realizada com algumas dificuldades pois a documentação e suporte disponíveis revelaram-se insuficientes. Muita da aprendizagem foi realizada recorrendo a sítios na Internet que abordavam tanto o Mondrian como o GeoMondrian. Ao longo da sua utilização deparou-se com alguns erros na medida em que a ferramenta não assumia algumas modificações realizadas, penso que devido ao uso de informação em cache resultante de execuções anteriores, a solução passava pelo reinício da ferramenta. A aprendizagem da ferramenta foi efetuada tentando colocar alguns exemplos recolhidos na internet em funcionamento, tendo sido possível apenas no típico exemplo do *geofoodmart*. O exemplo funciona mas sem possibilidade de visualização de dados em mapas, aliás foram encontradas algumas apresentações deste exemplo onde era possível visualizar informação extraída da DW em mapas, mas nenhum tutorial de como colocar a solução a funcionar. A ferramenta Workbench que deveria ajudar o utilizador a construir o cubo de forma gráfica, revelou ser muito lenta e ineficaz na publicação do cubo no motor do GeoMondrian. Não se considera que a utilização da ferramenta Workbench tenha sido uma mais-valia para o projeto. O JPivot, que permite a visualização das variáveis em análise, está integrado no GeoMondrian. A abordagem à ferramenta foi feita usando o exemplo *geofoodmart*, permitindo entender como colocar o JPivot a ler dados de qualquer DW. Aquando de erros de processamento do cubo, as mensagens de erro emitidas pelo JPivot deveriam ser menos ambíguas e mais precisas, raras foram as vezes que foi detetada a razão do erro através da leitura dos erros devolvidos pelo JPivot tanto pelo browser como pelos logs do TomCat. Os manuais e wikis referentes ao GeoMondrian não se revelaram adequados. Considero que para a solução ter mais sucesso no futuro, os seus contribuidores deveriam dar mais atenção a este facto.

Apesar do contratempo que impediu a utilização da ferramenta SOLAPLayers devido à sua descontinuidade, a visualização da distribuição

espacial das variáveis em análise foi conseguida recorrendo à construção de um protótipo utilizando a API Javascript do OpenLayers.

A escolha de utilização de software gratuito revelou-se vantajosa. A razão mais óbvia tem a ver com o custo da construção do sistema que foi nulo. A interação entre as diversas ferramentas foi conseguida de ponta a ponta, provando-se que apesar dos dados serem manuseados por diversas ferramentas, a integridade dos mesmos não foi posta em causa. A independência face à escolha de software Open Source revelou ser aposta acertada, dando como exemplo o facto de apesar do SOLAPLayers ter sido descontinuado, facilmente se optou por outra solução usando a API Javascript do Openlayers. Para tal foi unicamente necessário adaptar o código fonte do JPivot de forma a incluir esta API. O que demonstra que a utilização deste tipo de ferramentas permite uma personalização mais elevada.

Olhando para o resultado final do projeto, considera-se que a análise das variáveis espaciais, seria um dos aspetos a melhorar, nomeadamente a sua interface. As melhorias passariam pela interface permitir a renderização automática dos dados obtidos pelo JPivot, evitando ter-se que pressionar o botão para invocar a API do Openlayers e também incluir a possibilidade de maior flexibilidade na definição das legendas associadas às variáveis analisadas.

4 Conclusão e Trabalho Futuro

Pretendeu-se com este trabalho contribuir para a utilização de ferramentas *Open Source* na obtenção de um recurso credível e sem custos que pudesse potenciar uma análise mais crítica sobre o estudo realizado pela Escola Superior de Saúde do Instituto Politécnico de Viana do Castelo em articulação com o Observatório de Saúde do Gabinete de Cidade Saudável.

Neste contexto, este projeto centrou-se no estudo, análise e experimentação de Business Intelligence (BI) sobre informação geográfica. No que se refere à experimentação, foram utilizadas várias ferramentas *Open Source* GeoKettle, GeoMondrian, JPivot e numa fase inicial SOLAPLayers que acabou por ser substituída pelo OpenLayers, para potenciar uma análise que contemplasse a componente geográfica dos dados referentes a um estudo epidemiológico relativo ao perfil de saúde de um concelho de Portugal.

As dificuldades encontradas ao longo do desenvolvimento do projeto, foram na sua maioria causadas pela falta de documentação e suporte prestados pelas entidades que desenvolvem os diversos softwares utilizados. Estas dificuldades foram superadas com muita persistência, tanto recorrendo a técnicas de experimentação tentativa/erro como recorrendo à comunidade em geral consultando diversos sítios não oficiais na internet onde a utilização destas ferramentas é debatida.

É de realçar que apesar destes contratempos foi demonstrado neste projeto que é possível criar um sistema SOLAP recorrendo apenas a ferramentas *Open Source*, tornando assim esta tecnologia mais acessível a empresas com poucos recursos financeiros.

Relativamente ao trabalho futuro, penso que seria interessante adicionar uma dimensão “tempo” à data warehouse de forma a potenciar comparações entre as diversas datas de realização de inquéritos caso estes sejam novamente realizados. Outro aspeto que poderia ser aprofundado é a passagem de informação entre os dados geográficos obtidos no JPivot para um sistema cartográfico. Neste projeto realizou-se um protótipo de forma a

demonstrar que seria possível a visualização de polígonos obtidos na datawarehouse num mapa, mas por falta de tempo o aspeto gráfico e funcional não foi muito desenvolvido, ficando assim como trabalho futuro.

Considera-se que os resultados obtidos, ainda que baseados num protótipo, evidenciam a vantagem de introduzir a componente geográfica dos dados na sua análise, pelo facto de permitirem a visualização de resultados sobre a forma de mapas, oferecendo esta representação gráfica a possibilidade de realizar uma leitura espacial dos dados, permitindo, por exemplo, a visualização espacial das variáveis em análise com diferentes níveis de agregação.

5 Referências Bibliográficas

1KeyData, s.d. [Online]

Available at: <http://www.1keydata.com/datawarehousing/inmon-kimball.html>

[Acedido em 20 04 2012].

Azemoy, Smith e Sicherman, 1981. s.l.:s.n.

Bernardino, M. T. e. J., 2011. Open Source Business Intelligence Tools for SMEs. *Departamento Engenharia Informática e Sistemas IPC - Instituto Superior de Engenharia de Coimbra*.

Dr Michael de Smith e Prof Paul Longley, U. C. L. e. P. M. G. U. S. B., 2013. *Geospatial Analysis - A comprehensive guide*, <http://www.spatialanalysisonline.com>: s.n.

Etienne Dubé, T. B., 2009. *Geospatial BI with FOSS: an introduction to GeoMondrian and Spatialytics*. Laval University, Quebec, Canada, s.n.

Ficher, D. A. G. e. J. L., s.d. *Open source for Enterprise Geografic Information Systems*. s.l.:s.n.

GHOSH, R. A., 2006. *Economic impact of open source software on innovation and the competitiveness of the information and communication technologies (ICT) sector in the EU*.

Maastricht: UNU-MERIT. [Online]

Available at: <http://www.flossimpact.eu>

[Acedido em 12 04 2012].

Gnu, s.d. [Online]

Available at: <http://www.gnu.org>

[Acedido em 03 05 2012].

Klaus Bohm, A. M.-B. e. D. F., 2011. Geo Visual Analytics in the Public Health Sector. *University of Applied Sciences Mainz*.

Koshkarirov, s.d. 1989. s.l.:s.n.

LI Guangyu, Z. K. W. J. S. L. W. Q. Q. Y., 2010. Implementation of Publication System of Geological Data Using Open Source GIS. *Graduate University of Chinese Academy of Sciences*.

Livre, A. d. E., s.d. [Online]

Available at: <http://ensinolivre.pt/quantum-gis-qgis-sistema-de-informa%C3%A7%C3%A3o-geogr%C3%A1fica-livre-e-simplificado>

[Acedido em 20 10 2012].

OGC, 2011. *OGC White Paper – Geospatial Business Intelligence (GeoBI) [2011]: OGC 09-044r2*. s.l., OGC.

OpenSource, s.d. [Online]

Available at: <http://www.opensource.org/docs/definition.php>

[Acedido em 03 04 2012].

OSGeo, s.d. [Online]

Available at: www.osgeo.org

Pentaho, s.d. *Mondrian Workbench*. [Online]

Available at: <http://mondrian.pentaho.com/documentation/workbench.php>

[Acedido em 12 06 2012].

Pinto, I., s.d. *www.idcplp.net*. [Online]

Available at: http://www.idcplp.net/archive/doc/georrefIntroducaoSIG_InesPinto.pdf

[Acedido em 2012 03 01].

SABINO, V. & KON, F., 2009. *Licenças de Software Livre, História e Características. Relatório Técnico..* [Online]

Available at: <http://ccsl.ime.usp.br/files/relatorio-licencas.pdf>

[Acedido em 25 04 2012].

Spatialytics, s.d. *Geomondrian*. [Online]

Available at: <http://www.spatialytics.org/projects/geomondrian/>

[Acedido em 03 05 2012].

Tomaszewski, B., 2009. *Emerging Applications and Challenges for Geovisual Analytics Research*. [Online]

Available at: <http://rc.rit.edu/docs/rcss-tomaspresentation.pdf>

[Acedido em 05 04 2012].

Wang Lianbo, J. G. Z. H., 2010. Management Information System for Liaohe River Estuary Wetland Based on Open Source WebGIS. *College of Information Science and Engineering Ocean University of China*.

Webgis, s.d. [Online]

Available at: <http://www.webgis.com.br/postgis/>

[Acedido em 20 10 2012].

Wen Wang, H. L., 2010. Design and Construction of Environmental Monitoring Data Public System Based on WebGIS Platform. *School of Environment and Natural Resources Renmin University of China*.

Wikipédia, 2013. *Software livre*. [Online]

Available at: http://pt.wikipedia.org/wiki/Software_livre

[Acedido em 1 07 2013].

Wikipedia, s.d. [Online]

Available at: http://pt.wikipedia.org/wiki/Apache_Tomcat

[Acedido em 25 09 2012].

Wikipedia, s.d. *Wikipedia*. [Online]

Available at:

http://pt.wikipedia.org/wiki/Sistema_de_informa%C3%A7%C3%A3o_geogr%C3%A1

[Acedido em 01 03 2012].

6 Anexos

6.1.1 Alterações introduzidas ao testpage.jsp

Apresentam-se as alterações introduzidas ao ficheiro testpage.jsp. Estas alterações foram implementadas usando API Javascript do OpenLayers, permitindo a visualização num mapa, da dimensão Geo devolvida pela renderização efetuada pelo JPivot ao cubo.

```
<script type="text/javascript" src="ext-3.4.0/adaptor/ext/ext-base.js"></script>
<script type="text/javascript" src="ext-3.4.0/ext-all.js"></script>
<link rel="stylesheet" type="text/css" href="ext-3.4.0/resources/css/ext-all.css" />
<script type="text/javascript" src="OpenLayers-2.11/OpenLayers.js"></script>
<script type="text/javascript" src="GeoExt/lib/GeoExt.js"></script>
<link rel="stylesheet" type="text/css" href="GeoExt/resources/css/popup.css" />

// DADOS DE TESTE - simula os dados obtidos do MDX
// é um array de elementos que têm dois atributos:
//   - value com o valor que irá ser utilizado na simbologia de acordo com as regras definidas (a seguir)
//   - geometry com a string WKT

var dados = [];

// REGRAS DE FORMATAÇÃO DE TESTE
// é um array de elementos que pode ter até 4 atributos:
//   - liminf: valor inferior para a utilização do símbolo (opcional)
//   - limsup: valor superior para a utilização do símbolo (opcional)
//   - legenda: texto para a legenda
//   - style: definição do símbolo a utilizar
// Na definição do estilo (atributo style) é possível definir os atributos relacionados com
propriedades de representação de polígonos
// Podem encontrar a lista de todos os atributos no fim da página em
http://docs.openlayers.org/library/feature\_styling.html
// 1 - Objecto com uma única formatação para todos os elementos
var regra1 = [ { 'legenda': 'Dados que verificam todas as condições...',
                 'style': { fillColor: "#FF0000",
                           fillOpacity: 0.4, // de 0 a 1
                           strokeColor: "#00FF00",
                           // strokeOpacity: 1
                           //strokeWidth: 3,
                           strokeDashstyle: 'solid' // as opções são
                           dot, dash, dashdot, longdash, longdashdot
                         }
               }
];

// 2 - Objecto com várias regras de formatação para todos os elementos, definidas através dos
atributos liminf e limsup
// sem atributo liminf definido - formatação para valores inferiores de limsup
```

```
// sem atributo limsup definido - formatação para valores superiores a liminf
// sem atributo liminf e limsup definido - formatação para todos os valores (como caso do exemplo
anterior)
// outros casos - formatação para valores menores ou iguais a limsup e maiores ou iguais a liminf
var regra2 = [ { 'limsup': 10,
                 'legenda':'valores < 10',
                 'style': { fillColor: "#FF0000", fillOpacity: 0.8
                           }
               },
               { 'liminf':11, 'limsup': 50,
                 'legenda':'valores entre 11 e 50',
                 'style': { fillColor: "#00FF00", fillOpacity: 0.8
                           }
               },
               { 'liminf': 51,
                 'legenda':'valores > 51',
                 'style': { fillColor: "#0000FF", fillOpacity: 0.8
                           }
               }
             ];

// Funções de teste - invocadas ao clicar nos botões
// só tem de invocar o carregaMapa com um array de dados e outro de regras de acordo com o formato
anterior
function geraRegra1() {
    carregaMapa(dados, regra1);
}

function geraRegra2() {
    carregaMapa(dados, regra2);
}

// INICIO DO CÓDIGO GERAL PARA UTILIZAR NO VOSSO PROJECTO
// Definição do limite geográfico inicial do mapa
// limite para portugal
var maxBounds = new OpenLayers.Bounds(-9.6,36.8,-6.12,42.1 );

//var maxBounds = new OpenLayers.Bounds(-130000,-310000,170000,280000 );

// limite para o Concelho -> Escolher este ou o anterior
var maxBoundsVC = new OpenLayers.Bounds(-8.9,41.3,-8.06,42.2 );
// Inicio de código geral
Ext.BLANK_IMAGE_URL = "ext-3.4.0/resources/images/default/s.gif";
var mapa;
var lyrVector; // Variavel com tema para acrescentar poligonos

Ext.onReady(function() {
    // definição das opções do mapa
    var options = {
        controls: [new OpenLayers.Control.Navigation()],
        maxExtent: maxBounds,
        maxResolution: "auto",
        projection: "EPSG:4258", //"EPSG:3763",
        units: 'm'
    }
```

```
// ,allOverlays: true // Não existem "Base layers"

};
// Criar o mapa
mapa = new OpenLayers.Map();
mapa.setOptions(options);

// Criar um tema geográfico de fundo com base em distritos obtido diretamente do site do
IGP

// Definição da simbologia recorrendo ao formato SLD
theSLD_BODY= '<?xml version="1.0" encoding="utf-8"?>' +
  '<StyledLayerDescriptor version="1.0.0" xmlns="http://www.opengis.net/sld"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sld
http://schemas.opengeospatial.net/sld/1.0.0/StyledLayerDescriptor.xsd">' +
  '<NamedLayer><Name>NUT1-2012</Name>'+
  '<UserStyle><FeatureTypeStyle><Rule>'+
  '<PolygonSymbolizer><Fill><CssParameter name="fill">#5599DD</CssParameter></Fill>'+
  '<Stroke><CssParameter name="stroke">#000000</CssParameter></Stroke>'+

'</PolygonSymbolizer></Rule></FeatureTypeStyle></UserStyle></NamedLayer></StyledLayerDescriptor>';

var lyrIGP = new OpenLayers.Layer.WMS("Distritos",
  "http://mapas.igeo.pt/ows/caop/continente",
  {
    LAYERS: 'NUT1-2012',
    format: 'image/png',
    transparent: true,
    "SLD_BODY": theSLD_BODY
  },
  { isBaseLayer: true }
);

lyrIGP.setVisibility(false);
mapa.addLayer(lyrIGP);

// Criar layer para acrescentar os polgonos de teste
lyrVector = new OpenLayers.Layer.Vector("Vector Layer");
mapa.addLayer(lyrVector);

// Criar panel do mapa
var mapPanel = new GeoExt.MapPanel({
  id: 'mappanel',
  title: "Mapa",
  region: "center",
  //margins: '5 0 0 0',
  map: mapa
});

// criar panel para legenda
var westPanel = new Ext.Panel({
  id: 'legpanel',
  title: 'Legenda',
  region: 'west',
  //margins: '5 0 0 0',
  //cmargins: '5 5 0 0',
  width: 200,
  minSize: 100,
  maxSize: 350,
```

```
        collapsible: true,
        split: true
    });
    // Criar o layout global
    var mainPanel = new Ext.Panel({
        renderTo: "mainpanel",
        width: 800,
        height: 450,
        layout: "border",
        items: [mapPanel, westPanel]
    });

    mapa.addControl(new OpenLayers.Control.MousePosition());
    mapa.zoomToMaxExtent();

    mapa.setCenter(new OpenLayers.LonLat(-8.7, 41.7), 12); // Mapa centrado num dado ponto com
    um determinado nivel de zoom
    });

function carregaMapa(lista, regras) {
    // remove todos os elementos desenhados anteriormente caso existam
    lyrVector.removeAllFeatures();
    // Criar o objeto de regras para a legenda e controlo da simbologia no mapa
    var rules = criaRegras(regras);
    // criar estilo para a simbologia no mapa com base nas regras
    var style = new OpenLayers.Style();
    style.addRules(rules);
    // Atribuir o estilo ao mapa
    var stylemap = new OpenLayers.StyleMap({'default': style });
    lyrVector.styleMap = stylemap;
    // Criar objeto com capacidade para ler o formato WKT
    var formatador = new OpenLayers.Format.WKT();
    // Adicionar os elementos geográficos ao mapa
    for(var i=0; i<lista.length; i++) {
        var features = formatador.read(lista[i].geometry);
        // Criar uma objeto vector com o atributo value que será utilizado na simbologia de
        acordo com as regras
        var polyFeature = new
    OpenLayers.Feature.Vector(features.geometry,{'value':lista[i].value});
        // adicionar o objeto ao layer vector
        lyrVector.addFeatures(polyFeature);
    }
}

function criaRegras(regras) {
    rules = [];
    for(var i=0; i<regras.length; i++) {
        var filtro; // variavel para armazenar o filtro
        if (regras[i].liminf && regras[i].limsup) // Dois limites definidos
            filtro = new OpenLayers.Filter.Comparison({
                type: OpenLayers.Filter.Comparison.BETWEEN,
                property: "value",
                upperBoundary: regras[i].limsup,
                lowerBoundary: regras[i].liminf
            });
        else if (regras[i].liminf) // liminf definido
            filtro = new OpenLayers.Filter.Comparison({
                type: OpenLayers.Filter.Comparison.GREATER_THAN,
```

```
        property: "value",
        value: regras[i].liminf,
    });
    else if (regras[i].limsup) // limsup definido
        filtro = new OpenLayers.Filter.Comparison({
            type: OpenLayers.Filter.Comparison.LESS_THAN,
            property: "value",
            value: regras[i].limsup,
        });
    else
        filtro = null;
    rules.push(new OpenLayers.Rule({
        title: regras[i].legenda,
        filter: filtro,
        symbolizer: regras[i].style
    }));
}
// Criar e colocar a legenda
var legenda = new GeoExt.VectorLegend({
    //legendTitle: "Simbologia",
    showTitle: true,
    rules: rules,
    symbolType: "Polygon",
    selectOnClick: true
    //,renderTo: "legpanel"
});
var legpanel = Ext.getCmp('legpanel');
legpanel.removeAll();
legpanel.add(legenda);
legpanel.doLayout();
// devolver regras
return rules;
}

</script>
<title>GeoExt Example</title>

</head>
<body bgcolor=white>
<h2>Exemplo de utilização de WKT com sistema de coordenadas PT-TM06</h2>
    <br/>
    <input type="button" value="Uma única regra" onclick="geraRegra1();" />
    <input type="button" value="Várias regras" onclick="geraRegra2();" />
    <br/><br/>
    <div id="mainpanel"></div>
```

6.1.2 Ferramentas Comerciais

Apresenta-se uma lista de software GIS e de ferramentas e análise de vertente comercial (Dr Michael de Smith e Prof Paul Longley, 2013)

Produto	Fornecedor
---------	------------

Alteryx Brings Geographic Business Intelligence	Alteryx
Type: Geographic Business Intelligence	http://www.alteryx.com/
Alteryx provide the Windows Azure DataMarket with API services and the same datasets currently offered on the company's technology platform: Geosearch /Geocoding U.S. Census Data Current Year and Projected Demographic Data Consumer Expenditures Themed Map-Tile Service for Bing	

CubeWerx	CubeWerx
Type: BI solution	http://www.cubewerx.com
CubeWerx Suite is a complete, out-of-the-box infrastructure for managing, sharing and publishing geo-data on the Web. It includes all products listed on the left navigation panel.	

AgentSheets	AgentSheets Inc
Type: Geosimulation	http://www.agentsheets.com/
Simple commercial agent-based simulation package. An authoring tool that allows non-programmers to create agents with behaviors and missions, teach agents to react to information and process it in personalized ways, and combine agents to create sophisticated interactive simulations and models	

Arc2Earth	Arc2Earth
Type: GIS tools	http://www.arc2earth.com/
ArcGIS to Google Earth converter. Arc2Earth can export data in several different layouts. All of these layouts are useful in different ways from within Google Earth. Some are good for preserving the exact cartographic characteristics of the map and some are better for 3D extrusion and user interaction inside Google Earth. Any of these methods can also be combined together to form complex KML documents	

ArcGIS	ESRI Inc, Redlands, CA, USA
Type: GIS	http://www.esri.com/
General purpose, comprehensive, very extensive toolsets, vector focused with substantial raster support. Cross industry, Open Geospatial Consortium (OGC) compliant. Core "analysts" include: Geostatistical analyst: ESDA and geostatistics; Districting: free districting/re-districting utility; Tracking analyst: spatio-temporal tools; Spatial analyst: Surface/raster tools including basic interpolation; 3D Analyst: 3d analysis and visualisation; Network analyst: shortest path and related network analysis tools.	

AutoCAD Civil 3D	AutoDesk, San Rafael, USA
Type: Visualisation (2D and 3D)	http://usa.autodesk.com/civil-3d/
CAD/Engineering with Geospatial mapping and analysis	

Boundaryseer	Terraseer Inc, Ann Arbor, Mi, USA
Type: Specialised data analysis	http://www.terraseer.com/products_boundaryseer.php
Designed for the detection, description and analysis of geographic boundaries. It detects patterns and then tests them statistically	

CadCorp SIS	Cadcorp Ltd, Stevenage, UK
Type: GIS	http://www.cadcorp.com/products_geographical_information_systems/
CadCorp SIS (Spatial Information System) from UK. An integrated suite of desktop GIS, developer kits and mobile	

Didger	Golden Software Inc, Golden, CO, USA
Type: Specialised mapping	http://www.goldensoftware.com/products/didger/didger.shtml
Manual and automated digitising, georeferencing, reprojecting/coordinate conversion, and mosaicking software, with extensive import/import	

ER Mapper	ER Mapper, Woking, Surrey, UK
Type: Specialised mapping	http://www.ermapper.com/
(purchased by Leica in 2007). Provider of geospatial image processing, compression and distribution software. Handle very large image files and databases (terabytes)	

ERDAS Imagine	ERDAS, USA
Type: GIS	http://www.erdas.com/
A suite of software tools designed specifically to process geospatial imagery - strong focus on spectral analysis and high quality classification. Many add-on modules, including "Image Analysis for ArcGIS", which enables data to be processed and loaded straight into an ArcGIS geodatabase.	

FME	Safe Software, Surrey, BC, Canada
Type: Specialised mapping	http://www.safe.com
GIS Data Translator: FME supports fast and easy conversion between over 50 projections and over 6000 predefined coordinate systems. Spatial data extract,	

transform and load (ETL) tools.

GeoExpress	Lizardtech
Type: Image handling	http://www.lizardtech.com/
Geospatial Image Processing and Data compression (MrSid, DjVu). GeoExpress 6.1 combines the strengths of the industry-standard image compression format, MrSID, with the ISO-standard JPEG 2000, and integrates powerful image editing and manipulation tools to bring you the most advanced geospatial imaging software on the market - includes tools for reprojection, advanced color balancing, area of interest encoding and multiresolution mosaicking	

Geographic Calculator	Blue Marble Geographics, Maine, USA
Type: GIS tools	http://www.blumarblegeo.com
One of a set of Bluemarble's geographic utility software, which includes coordinate conversion, image re-projection and related utilities and code libraries	

Geostat	Geostat Systems International Inc, Quebec, Canada
Type: Geostatistical analysis	http://www.geostat.com/English/products.htm
Geostatistical software (and related products and services) aimed at the mining industry.	

GS+	Gamma Design Software, Plainwell, MI, USA
Type: Geostatistical analysis	http://gammadesign.com/
Geostatistical analysis program for desktop systems. Introduced in 1988 as the first integrated geostatistics program for the PC. GS+ offers all components from semivariance analysis through kriging, conditional simulation and	

mapping in an integrated package.

Idrisi	Clark Labs, Clark University Worcester, MA, USA
Type: GIS	http://www.clarklabs.org/
A sophisticated GIS and Image Processing software solution that includes over 200 modules for the analysis and display of digital spatial information. Special facilities are included for environmental modeling and natural resource management, including change and time series analysis, land change prediction, multi-criteria and multi-objective decision support, uncertainty analysis and simulation modeling. TIN interpolation, Kriging and conditional simulation are also offered. Raster-based product, especially for environmental sciences. Remote sensing, land management. See also, Land Change Modeller (LCM) for Idrisi and ArcGIS	

Instant Atlas	Geowise, Edinburgh, UK (owned by ESRI inc)
Type: Specilaised mapping	http://www.instantatlas.com
Web-based analytical mapping and reporting, also with desktop option. Strong focus on statistical mapping for health and government sectors. ESRI business associate	

ISATIS	Geovariances, Avon, France
Type: Geostatistical analysis	http://www.geovariances.com/
Geostatistical software for the Earth sciences - aimed especially at Oil, Gas and Mining sectors.	

Manifold	Manifold Inc, Carson City, NV, USA
Type: GIS	http://www.manifold.net/
Manifold is described by its suppliers as a "word processor for maps". It is also described as a "database system for maps" that will allow you to embed	

database information into a map so that the map can be used as a visual interface into the data. General purpose, very extensive toolsets, vector focused with raster support. Cross industry, OGC compliant. Many analytical functions are built-in via the ViewBots facility and the Transform operator tool - this includes basic operations such as buffer and clip, but also more complex operations such as spline smoothing and Voronoi region computation. A range of surface analysis operations (e.g. map algebra, visibility analysis, interpolation etc.) plus network analysis (optimal routes, drive time zones etc) are included

MapInfo	Mapinfo Inc, troy, NY, USA
Type: GIS	http://www.pbinsight.com/welcome/mapinfo/
General purpose, vector focused with raster support. MapInfo Professional is a powerful Microsoft Windows-based mapping application that enables business analysts and GIS professionals to easily visualize the relationships between data and geography. Cross industry/Marketing; HotSpot Detective (J Ratcliffe) for crime analysis. Tools provided include basic operations (e.g. buffering, distance and area computations etc); re-districting tools, a range of surface operations (e.g. interpolation, TIN operations, viewshed analysis etc.)	

MapViewer	Golden Software Inc., Golden, CO, USA
Type: GIS	http://www.goldensoftware.com/products/mapviewer/mapviewer.shtml
A simple but powerful mapping and spatial analysis tool that facilitates production of publication-quality thematic maps	

R2V	Able Software Corp, Lexington, MA, USA
Type: GIS tools	http://www.ablesw.com/r2v/index.html
Raster to Vector conversion software	

Software listing	
Type: GIS tools	http://ecat.giscale.com/index.php
Listing of a variety of GIS related software	

TatukGIS	TatukGIS, Gdynia, Poland
Type: GIS tools	http://www.tatukgis.com/
GIS software objects for application development (SDKs). TatukGIS Company designs, develops, licenses, markets, and supports Geographic Information System (GIS) focused software products and software development tools for the creation of custom GIS applications and solutions. The leading product is the TatukGIS ‘Developer Kernel’, a comprehensive GIS development toolkit (SDK) that is licensed to GIS software solution developers in approximately 50 countries.	

6.1.3 Ferramentas Baseadas em Tecnologia Livre

Apresenta-se uma lista de software GIS e de ferramentas e análise baseadas em Tecnologia Livre (Dr Michael de Smith e Prof Paul Longley, 2013)

Produto	Fornecedor
----------------	-------------------

SpagoBI	
Type: BI	http://www.spagoworld.org/xwiki/bin/view/SpagoBI/
The SpagoBI project is a free software/open source initiative by the SpagoBI Competency Center of Engineering Group, one of the top 10 European Information Technology companies. It aims to realize the most complete 100% open source business intelligence suite, aggregating developers, integrators, companies, users and passionate people in an open community.	

OpenI	
Type: BI	http://openi.org/
OpenI-Jasper provides a plug-in to replace JasperSoft OLAP UI, switching the old JPivot interface with a cleaner UI to view OLAP data. OpenI Server is the original OpenI open source project. It is a Business Intelligence web application for on-demand or SaaS deployments.	

Pentaho	
Type: BI	http://www.pentaho.com
Pentaho Business Analytics is ready to run out of the box as a stand-alone BI application, providing highly interactive and easy to use self-service web interfaces for business users to access, visualize, and analyze business information	

JasperSoft	
Type: BI	http://www.jaspersoft.com/
JasperReports Server is a powerful, yet flexible and lightweight reporting server. Generate, organize, secure, and deliver interactive reports and dashboards to users. Allow non-technical users to build their own reports and dashboards.	

Palo	
Type: BI	http://www.palo.net/
Palo Suite combines all core applications — OLAP Server, Palo Web, Palo ETL Server and Palo for Excel — into one comprehensive and customisable Business Intelligence platform. The platform is completely based on Open Source products representing a high-end Business Intelligence solution which is available entirely free of any license fees.	

Vanilla	
Type: BI	http://forge.bpm-

	conceil.com/projects/vanilla/
Vanilla is the Open Source Business Intelligence platform which enables you to deploy whole business intelligence processes such as DWH, Workflow, ETL, business layers, reports, dashboards and cubes.	

ArcExplorer	ESRI Inc, Redlands, CA, USA
Type: Viewer	http://www.esri.com/software/arcexplorer
ArcExplorer--GIS Data Viewer - a lightweight GIS data viewer written in Java that is used to perform basic GIS functions (e.g., view, navigate, and query). It is a downloadable application that operates in a stand-alone environment	

ESTAT	Penn State
Type: Exploratory data analysis	http://gis.cancer.gov/nci/spatial.html
Exploratory Spatio-Temporal Analysis Toolkit, a Java-based implementation of several of the ESDA tools provided within GeoVista augmented by linked time-series plots	

FDO	Open Source team,/Autodesk
Type: GIS tools	http://fdo.osgeo.org/
FDO (Feature, Data, Object) Data Access Technology is an API for manipulating, defining and analyzing geospatial information regardless of where it is stored. FDO uses a provider-based model for supporting a variety of geospatial data sources, where each provider typically supports a particular data format or data store	

GeoDa	Spatial Analysis Laboratory, Univ of Conneticut, USA
Type: Exploratory data analysis	http://geodacenter.asu.edu/software
Exploratory spatial data analysis, vector (L Anselin). GeoDa is the latest	

incarnation of a collection of software tools designed to implement techniques for exploratory spatial data analysis (ESDA) on lattice data.¹ It is intended to provide a user friendly and graphical interface to methods of descriptive spatial data analysis, such as global and local (LISA) autocorrelation statistics and indicators of spatial outliers, plus some more advanced regression analysis facilities.

Geotools	Collaborative effort
Type: GIS tools	http://geotools.codehaus.org/
Open Source GIS Java toolset providing implementations of many Open Geospatial Consortium (OGC) specifications as they are developed. GeoTools is also associated with the GeoAPI project that creates geospatial, Java interfaces	

MapGuide	
Type: GIS	http://mapguide.osgeo.org/
MapGuide Open Source is a web-based platform that enables users to quickly develop and deploy web mapping applications and geospatial web services. MapGuide features an interactive viewer that includes support for feature selection, property inspection, map tips, and operations such as buffer, select within, and measure. MapGuide includes an XML database for managing content, and supports most popular geospatial file formats, databases, and standards	

Mapserver	Univ of Minesotta, USA
Type: Specialised mapping	http://mapserver.org/
MapServer is an Open Source development environment for building spatially-enabled internet applications. MapServer is not a fullfeatured GIS system, nor does it aspire to be. Instead, MapServer excels at rendering spatial data (maps, images, and vector data) for the web. A substantial 'gallery' of applications and sites using Mapserver is accessible via: http://mapserver.gis.umn.edu/gallery	

OpenMap	BBN Technologies, Cambridge, MA, USA
Type: GIS tools	http://openmap.bbn.com/
Open source, Java Beans based geospatial mapping toolset	

Open SourceGIS	Open SourceGIS
Type: GIS tools	http://Open Sourcegis.org/
Assorted open source GIS software list, including almost 250 sites and software utilities for spatial data processing	

PostGIS	Refractions research, Victoria, BC, Canada
Type: GIS	http://postgis.refractions.net/
PostGIS adds support for geographic objects to the PostgreSQL object-relational database. In effect, PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS)	