



INSTITUTO POLITÉCNICO  
DE VIANA DO CASTELO

Marcelo dos Santos Videira

# **Sistemas de Processamento Analítico aplicado a Informação Geográfica**

Tecnologia e Gestão de Sistemas de Informação

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

Julho de 2013



## Agradecimentos

Em primeiro lugar agradeço ao Professor Doutor Jorge Ribeiro e Professor Doutor Pedro Castro pela orientação e pelo apoio ao longo da realização do caso de teste.

Agradeço também ao colega Silvestre Malta, pois foi um grande apoio ao longo da realização do projecto.

Por fim agradeço em especial aos meus pais e avó materna pela força e pelo incentivo, nos momentos mais difíceis.



## RESUMO

Ao longo dos tempos, é armazenado nos Sistemas de Informação um conjunto diversificado de dados sobre os quais as ferramentas de Business Intelligence permitem auxiliar os agentes de decisão nas suas tarefas diárias de tomada à decisão. Tipicamente uma das formas de solucionar a integração dos dados de diferentes sistemas ou aplicações é o desenvolvimento e implementação de Sistemas de Data Warehousing designados (SDW). Por sua vez, os sistemas de processamento analítico, vulgarmente definidos como sistemas On-line Analytical Processing (OLAP), operam sobre a informação dos repositórios dos SDW e permitem alargar significativamente os horizontes de análise dos agentes de decisão das organizações, ao disponibilizarem meios muito rápidos de análise sobre várias áreas de atividade das organizações. Estes sistemas assentam sobre a informação armazenada nos SDW, pré-calculando e processando todas as combinações dos agrupamentos de cada entidade dos SDW, e materializando-as por fim, em Estruturas Multidimensionais de Dados, que são designados de Cubos.

Os Sistemas de Informação Geográfica são conhecidos como um conjunto organizado de *hardware*, software, dados geográficos e pessoas, destinados a eficientemente obter, armazenar, atualizar, manipular, analisar e exibir todas as formas de informação referenciada geograficamente. Pela sua natureza, diversidade e potencialidade, estes sistemas tem vindo a ser reconhecidos como um meio expedito e útil de análise de dados, permitindo aos agentes de decisão complementarem as suas análises sobre os tradicionais dados alfanuméricos com informação georreferenciada, permitindo-lhes tirar partido da componente geográfica através das operações de análise espacial.

Paralelamente ao aparecimento dos dados georreferenciados, assiste-se a uma evolução nas ferramentas que permitem trabalhar com este tipo dados, para disponibilizar a informação numa janela de oportunidade útil para que os agentes de decisão. Por outro lado, tem crescido o interesse e disponibilização de novas funcionalidades quer por parte das empresas proprietárias de desenvolvimento de software, quer por grupos que se orientam pelo desenvolvimento de software de open source.

Neste contexto, este trabalho centra-se no estudo, análise e construção de um sistema de Business Intelligence usando informação geográfica desenvolvido com recurso a ferramentas Open Source. O caso de estudo utiliza informação real dos

censos 2011 para Portugal Continental. O caso é fundamentalmente uma prova de conceito da possibilidade de criar este tipo de aplicações com ferramentas Open Source, tendo como foco o desempenho das ferramentas OLAP.

Com o desenvolvimento deste trabalho é possível apresentar informação num mapa, permitindo uma representação espacial dos dados, como indicadores sobre população edifícios, entre outros, e recorrer às tradicionais operações de roll-up e drill-down amplamente reconhecidas nas ferramentas OLAP (e sistemas de Business Intelligence) quando aplicados a dados alfanuméricos.

**Palavras-Chave:** Sistemas de Suporte à Decisão, Sistemas de Processamento Analítico; Informação Geográfica, Sistemas de Informação Geográfica.

## Abstract

In the last decades, the Information Systems stores a diverse set of data used by Business Intelligence tools which allows decision makers to complement their information in their daily decisions. Typically one of the ways to solve the integration of data from different systems or applications is by developing and implementing Data Warehousing Systems (SDW). On another way, the analytical processing systems, commonly defined as Online Analytical Processing Systems (OLAP) operate on SDW information repositories and significantly extend the horizons of decision-makers analysis in order to make available means for very fast analysis on various areas of activity of organizations. These systems rely on the information stored in SDW, pre-calculating and processing all combinations of group-bys of each entity of the SDW, and materializing them finally in Multidimensional Data Structures, which are commonly called Cubes.

The Geographic Information Systems are known as an organized set of hardware, software, geographic data and people, designed to efficiently obtain, store, update, manipulate, analyze and display all forms of referenced geographically information. By their nature, diversity and capability, these systems have been recognized as a useful and expedient means of data analysis, enabling decision-makers to complement their analysis over traditional alphanumeric data with data that is georeferenced information allowing them take advantage of spatial data operations.

In parallel developments in terms of the implementation of such systems, one of the observations is the large volume of data and computational process that is required to provide the information in a useful opportunity window for the decision makers in order to base their analyzes on this type of georeferenced data. On the other hand, has been growing the interest and availability of new features by proprietary companies of software development, and by developer groups that guided their activity by the development of open source software.

In this context, this paper focuses on the study, analysis and construction of a Business Intelligence system using geographic information developed using Open Source tools. The case study uses real data from the census conducted in 2011 in Portugal. The case is essentially a proof of concept, in order to analyze the ability to create such applications with free tools, being focused on the performance of OLAP tools based on open source technology. With the development of this work it is possible to identify areas with more population, buildings, etc. displaying the information on a map using the traditional

operations of roll-up and drill-down tools widely recognized in (OLAP and Business Intelligence systems) widely used in alphanumeric data.

**Key-Words:** Decision Support Systems, Online Analytical Processing; Geographic Information Systems.



## ACRÓNIMOS

**BI** – Business Intelligence

**DM** – Data Mart

**DW** – Data Warehouse

**EMD** – Estruturas Multidimensionais de Dados

**EPGS** - European Petroleum Survey Group

**ERP** - Enterprise resource planning

**ETL** – Extract Transform Load

**MDX** - MultiDimensional eXpressions

**OGC** - Open Geospatial Consortium

**OLAP** – Online Analytical Processing

**SDW** – Sistemas de Data Warehousing

**SGBD** – Sistema Gestor de Base de Dados

**SIG** – Sistemas de Informação Geográfica

**SQL** - Structured Query Language

**SRID** - Spatial Reference System Identifier



# Índice

1. Introdução .....	1
1.1 Contextualização .....	1
1.2 Motivação e Objectivos.....	3
1.2.1 Motivação e Objectivos.....	3
1.3 Metodologia .....	4
1.2.4 Organização .....	5
2. Sistemas de Informação Geográfica (SIG) e Sistemas de Processamento Analítico (OLAP) .....	7
2.1 Introdução .....	7
2.2 Sistemas de Informação Geográfica .....	8
2.2.1 Ciclo de Vida .....	9
2.2.2 Representação dos dados .....	10
2.2.3 Armazenamento de Informação Geográfica .....	12
2.3 Sistemas de Processamento Analítico .....	16
2.3.1 Sistemas de Data Warehousing.....	17
2.3.3.Características dos Sistemas de Processamento Analítico .....	21
2.4 Reflexão Crítica.....	29
3. SIG e OLAP utilizando Tecnologias OpenSource .....	31
3.1 Introdução .....	31
3.2 Arquitectura Geral de Camadas.....	33
3.3 Estado da Arte.....	34
3.3.1 A Nível Académico .....	34
3.3.2. Implementações realizadas .....	38
3.3.3. Ferramentas SOLAP .....	39
3.4 Reflexão Crítica.....	42
4. Caso de Estudo .....	45
4.1 Introdução .....	45
4.2 Características da Informação.....	45
4.3 Arquitectura Geral e Ferramentas utilizadas .....	47
4.4. Tarefas realizadas .....	48
4.4.1 Instalação das ferramentas.....	48
4.4.2 Construção do Modelo Multidimensional.....	49

4.4.3 Construção do Processo de ETL .....	50
4.4.4 Construção do modelo SOLAP .....	58
4.4.5 Visualização Geográfica.....	62
4.5 Análise e Discussão .....	74
5. Conclusões .....	77
5.1 Introdução .....	77
5.2 Reflexão crítica.....	78
5.3 Trabalho Futuro.....	78
6. Referências Bibliográficas .....	81
Anexos .....	85
A.1 – Código javascript utilizado na construção da interface Web.....	85
A.2- Ficheiro de configuração Default do ficheiro postgresql.conf .....	89
A.3 - Ficheiro de configuração Development do ficheiro postgresql.conf.....	98
A.4 - Ficheiro de configuração Mixed do ficheiro postgresql.conf .....	107
A.5 - Ficheiro de configuração Dedicated do ficheiro postgresql.conf .....	116

## Índice de Figuras

Figura 1- Ciclo de vida de um Sistema de Informação Geográfico .....	9
Figura 2- Organização dos dados de um SIG (Fu & Sun, 2010).....	11
Figura 3- Caracterização da estrutura em Camadas de um SIG (Fu & Son, 2010).....	11
Figura 4- Representação num Sistema de Gestão de base de dados dos dados associados a um SIG.....	12
Figura 5 – Ilustração do conteúdo da tabela “spatial_ref_sys”.....	15
Figura 6 – Ilustração dos registos da tabela “Geometry columns”.....	16
Figura 7 - Ciclo de vida típico de um projeto de Suporte à Decisão (Microsoft) .....	20
Figura 8 - Esquemático tradicional de um Data Mart de um Data Warehouse usando uma estrutura em estrela. ....	23
Figura 9 - Esquemático tradicional de um Data Mart de um Data Warehouse usando uma estrutura em floco de neve.....	23
Figura 10- Processamento de Estruturas Multidimensionais de Dados.....	24
Figura 11 - Ilustração do processo de Extracção, Transformação e Carregamento fonte:( <a href="http://www.spatialytics.org/projects/geokettle/">http://www.spatialytics.org/projects/geokettle/</a> ).....	25
Figura 12 – Ilustração da estrutura geral de visualização de um cubo de dados (Siqueira, 2009) .....	26
Figura 13 – Exemplificação do resultado de operações OLAP usando dados alfanuméricos. Fonte: Exemplo GeoFoodMart do GeoMondrian .....	27
Figura 14 – Ilustração de uma visualização de Informação Geográfica com análise espacial de dados. Fonte: ( <a href="http://laser.ucs.inrs.ca/lmg/Carte_CanEU.jpg">http://laser.ucs.inrs.ca/lmg/Carte_CanEU.jpg</a> ) .....	32
Figura 15 - Arquitectura de um Data Warehouse espacial Adaptado: (Tripathy et al., 2010).....	33
Figura 16 – Esquema geral associado aos Sistemas OLAP com e sem tratamento de informação geográfica (2011).....	34
Figura 17 – Exemplificação de hierarquias de visualização de informação geográfica.36	
Figura 18 – Exemplificação do resultado de operações OLAP usando dados com informação geográfica (Silva, 2010).....	37
Figura 19 - Exemplo de um SIG. (Wildlife).....	39
Figura 20 – Ilustração de uma ferramenta OLAP (SpagoBI ) a operar sobre Informação Geográfica.....	40
Figura 21 - Exemplo de análise alfanumérica no Pentaho.....	41
Figura 22 – Tipo de informação disponibilizada pelo INE.....	46
Figura 23 - Características da máquina virtual.....	47
Figura 24 – Esquema da arquitectura geral do sistema.....	48
Figura 25 - Modelo Multidimensional do caso de estudo.....	50
Figura 26 - Carregar ShapeFile no Quantum GIS.....	51
Figura 27 - Caminho da opção “Dissolver”.....	51
Figura 28 - Escolha da coluna para efectuar agregação de dados.....	52
Figura 29 - Ficheiro CSV com dados alfanuméricos.....	52
Figura 30 – Ilustração da Interface da ferramenta GeoKettle.....	53
Figura 31 – Visualização do repositório (Base de Dados).....	53
Figura 32 – Ilustração do carregamento “Tranformations” do Repositório.....	54

Figura 33 - Exemplificação de uma “Transformation” .	54
Figura 34 – Definição do sistema de coordenadas.	55
Figura 35 - Exemplificação da operação de inserir dados numa tabela, com a especificação das colunas.	55
Figura 36 - Processo ETL final da dimensão Município.	56
Figura 37 - Exemplificação da “Tranformations” para carregar a tabela de factos.	57
Figura 38 - Pesquisa para efectuar a selecção de dados para tabela de factos.	57
Figura 39 - Ilustração das funcionalidades disponíveis no Workbench.	58
Figura 40 - Ilustração de um erro da execução de uma pesquisa MDX.	59
Figura 41 - Exemplificação da ligação entre o Workbench e o DW.	59
Figura 42 - Modelo da Estrutura do Cubo	60
Figura 43 - Detalhe da construção do Cubo	60
Figura 44 - Publicação do esquema no projecto.	62
Figura 45 - Ilustração do layout da aplicação.	63
Figura 46 - Apresentação dos dados alfanuméricos.	65
Figura 47 - Ilustração das operações de Drill-down Roll-Up.	66
Figura 48 - Apresentação de dados geométricos na Pivot table.	66
Figura 49 - Esquemático das operações Drill-Down e Roll-up sobre os dados	67
Figura 50 - Visualização ao nível das freguesias no concelho de Viana do Castelo	67
Figura 51 - Visualização ao nível das das secções da Freguesia de Perre	68
Figura 52 - Visualização ao nível das das secções da Freguesia de Perre	68
Figura 53 - Visualização geral do país.	69
Figura 54 - Análise do desempenho do Sistema – Análise para o Município de Viana do Castelo.	70
Figura 55 - Análise do desempenho do Sistema – Análise para o Município de Viana do Castelo e de Braga.	70
Figura 56 - Análise do desempenho do Sistema – Análise para o Município de Viana do Castelo, Braga e Lisboa.	71
Figura 57 - Análise do desempenho do Sistema – Análise para Portugal Continental.	71
Figura 58 - Analise das sessões activas na base dados.	72
Figura 59 - Instalação do plug-in para a configuração dos parâmetros do PostgreSQL	73
Figura 60 - Análise do desempenho do Sistema – Análise para Portugal Continental.	74

## **Índice de Tabelas**

Tabela 1 – Diferenças entre Sistemas Relacionais e Multidimensionais (Bédard et al, 2001).....	18
Tabela 2 – Diferenças entre Data Warehouse e Data Mart.....	18
Tabela 3 – Esquema XML do cubo .....	61
Tabela 4 - Definição dos links para análise dos dados. ....	63
Tabela 5 - Atribuição da pesquisa MDX ao parâmetro. ....	63
Tabela 6 - Renderizar a tabela com os dados. ....	64
Tabela 7 – Código necessário ao carregamento do Array para a visualização no mapa. .....	64
Tabela 8 – Código da pesquisa MDX para limitar os dados ao conselho de Viana do Castelo .....	65
Tabela 9 – Pesquisa SQL das sessões activas na base de dados.....	72
Tabela 10 - Comparação de tempos.....	72





# 1. Introdução

## 1.1 Contextualização

Tradicionalmente, os Sistemas de Informação transaccionais (por exemplo, um sistema ERP-Enterprise Resource Planning) implementados em grande parte das organizações tiveram a sua origem justificada pelas necessidades de suporte operacional e pelo armazenamento estruturado da sua informação. Porém, na generalidade dos casos, estes sistemas foram desenvolvidos autonomamente, não tendo sido construídos com base num modelo global da organização mas, pelo contrário, tipicamente em modelos de sistemas de suporte à informação heterogénea das várias actividades da organização. Contudo, com o aumento de volume de dados, a necessidade de armazenar o histórico e a conjugação de várias variáveis de negócio online, surge a necessidade de executar novas pesquisas mais complexas, sendo importante obter respostas rápidas para este tipo de requisitos por parte dos agentes de decisão. Uma das formas para solucionar a integração dos dados de diferentes sistemas é o desenvolvimento e implementação de Sistemas de Data Warehousing (SDW) na organização, sendo possível criar um modelo de dados global à organização em que a informação está centralizada em repositórios de dados históricos, integrados, validados e estruturados segundo as várias vertentes de negócio, como por exemplo, o tempo.

Tipicamente, existe um pré-processamento da informação para cada gestor da organização, de modo a que os relatórios de análise da informação de vários anos sejam apresentados de uma forma rápida e com informação coerente. Neste sentido, os sistemas de processamento analítico, vulgarmente definidos como sistemas *On-line Analytical Processing* (OLAP), operam sobre a informação dos repositórios dos Sistemas de Data Warehousing (SDW), que disponibilizam meios muito rápidos de análise sobre várias áreas de actividade das organizações. Estes sistemas assentam sobre a informação armazenada nos SDW, pré-calculando e processando todas as combinações dos agrupamentos de cada entidade existentes nos mesmos, materializando-as por fim em Estruturas Multidimensionais de Dados (EMD), vulgarmente designados de Cubos. Estes Cubos são tipicamente usados como base dos sistemas de Business Intelligence, assim como para sistemas de Descoberta de Conhecimento em Bases de Dados.

Por outro lado, nas últimas décadas, os Sistemas de Informação têm vindo a ser explorados e potenciados através da inclusão de determinadas características associadas à localização geográfica, quer de objectos como de pessoas. Estes sistemas, designados por Sistemas de Informação Geográfica (SIG) são definidos como um conjunto organizado de hardware, software, dados geográficos e pessoas, destinados a eficientemente obter, armazenar, actualizar, manipular, analisar e exibir todas as formas de informação referenciada geograficamente (Sutton et al., 2009). Torna-se assim necessário e extremamente útil disponibilizar aos agentes de decisão mecanismos simples e expeditos para analisarem e explorarem informação alfanumérica e espacial associada aos dados. Em suma, este tipo de sistemas, além de armazenar grandes volumes de dados, permite aos agentes de decisão obter respostas a perguntas do tipo “onde?”.

Numa outra vertente contextual, em termos de implementação de Sistemas de Informação, nos últimos anos, tem crescido o fenómeno de utilização de tecnologias livres e software open source. Este crescimento deve-se, entre outras razões, às questões associadas ao custo de licenciamento das tecnologias e ferramentas proprietárias, à existência de entidades promotoras da utilização, promoção e protecção do software open source (como por exemplo, a Free Software Foundation e a Open Source Initiative), assim como ao reconhecimento por parte de uma vasta comunidade de programadores e utilizadores da eficiência e abrangência da utilização deste tipo de software.

Em particular, onde se tem evidenciado um forte crescimento de novos módulos/funcionalidades usando software open source, é na área dos Sistemas de Informação Geográfica. Assim, tem-se constatado efectivamente um crescimento no desenvolvimento de inúmeros projectos de software open source, permitindo o acesso à tecnologia SIG pelo público em geral, em repositórios como o FreeGis

Neste contexto, de um ponto de vista, este projecto centra-se no estudo, análise e exploração dos conceitos associados aos Sistemas de Suporte à Decisão aplicados a Sistemas de Informação Geográfica. Em particular, pretende-se estudar e aplicar os conceitos e terminologias associadas aos Sistemas de Data Warehousing e Sistemas de Processamento Analítico quando aplicados a Sistemas de Informação Geográfica. Para atingir este fim, em termos de implementação, utilizamos ferramentas baseadas no conceito de software open source usando um caso de estudo real baseado em dados disponibilizados pelo Instituto Nacional de Estatística referentes aos censos 2011. Desta maneira, pretendemos estudar, analisar e testar algumas das particularidades dos típicos sistemas de processamento analítico amplamente utilizados pelas ferramentas de Suporte à Decisão com dados alfanúmericos, mas agora utilizando dados com componente georreferenciada.

## **1.2 Motivação e Objectivos**

### **1.2.1 Motivação e Objectivos**

Os agentes de decisão mais activos costumam frequentemente analisar todos os tipos de informação que têm à sua disposição, desenvolvendo processos de procura constante, em busca deste ou daquele indício, que lhes proporcione algumas vantagens no desenvolvimento das suas actividades e no posicionamento das suas organizações nos mercados em que estão inseridas. Neste contexto, os sistemas de informação geográfica têm vindo nos últimos anos a evidenciar um grande sucesso pois conseguem a unificação dos dados tradicionais com geográficos. Por sua vez, o sucesso do tratamento de grandes volumes de dados e das vantagens dos Sistemas de Processamento Analítico em Sistemas de Suporte à Decisão com informação georreferenciada, é uma nova janela de oportunidade. Neste contexto, a motivação deste trabalho centra-se no estudo das vantagens associadas à implementação de um Sistema de Processamento Analítico com a componente geográfica incorporada em detrimento de um Sistema de Processamento Analítico tradicional. Adicionalmente,

tendo em consideração a exploração deste tipo de sistemas aplicados a dados espaciais, pretendemos estudar, avaliar e implementar um sistema protótipo baseado em software open source para constatar os desafios e implicações do actual estado da arte.

Neste contexto, esta dissertação centra-se no estudo sobre os conceitos associados aos sistemas de processamento de estruturas multidimensionais aplicadas a temáticas assentes em sistemas de gestão de bases de dados geográficos. Neste sentido, pretende-se que este estudo seja exploratório, não procurando validar relações de causalidade nem verificar uma teoria. Pretende-se antes, identificar preocupações, dificuldades e hesitações na criação de sistemas que tirem partido dos tradicionais Sistemas de Processamento Analítico e dos Sistemas de Informação Geográfica. Assim, além do estudo sobre os conceitos dos sistemas de Data warehousing aplicados à informação georreferenciada, o objectivo específico centra-se na criação de um Sistema Protótipo de Data Warehousing (para armazenamento de dados) e de um Sistema de Processamento Analítico baseado em tecnologia open source para explorar a informação associada a um Sistema de Informação Geográfico, em que a informação é disponibilizada pelo Instituto Nacional de Estatística. Pretende-se também explorar as potencialidades deste tipo de sistemas no apoio à decisão, assim como analisar os constrangimentos e realidades dessa implementação usando um caso real de aplicabilidade.

### **1.3 Metodologia**

Para realizar este trabalho foram delineadas várias tarefas, tendo o estudo sido baseado na seguinte metodologia de trabalho:

1. Efectuar um estudo sobre a problemática do processamento e manutenção de Estruturas Multidimensionais em ambientes de processamento analítico (OLAP);
2. Efectuar uma análise detalhada das técnicas de processamento de Estruturas Multidimensionais;
3. Efectuar um estudo sobre os conceitos, terminologias e ferramentas associadas aos Sistemas de Informação Geográficos;
4. Analisar o estado da arte dos Sistemas de Data Warehouse e dos sistemas OLAP quando aplicados aos Sistemas de Informação Geográfica; assim como estudar as suas características e particularidades;

5. Implementar um sistema protótipo, no sentido de criar uma estrutura de suporte ao processamento OLAP aplicado a Sistemas de Informação Geográfica;
6. Explorar e avaliar os resultados obtidos da aplicabilidade do sistema protótipo.

#### **1.2.4 Organização**

Esta dissertação encontra-se estruturada em cinco capítulos. No primeiro apresenta-se uma contextualização a nível do enquadramento do trabalho, da motivação e dos seus objectivos. No segundo capítulo apresenta-se os conceitos e terminologias associados à Informação Geográfica, aos Sistemas de Informação Geográfica e aos Sistemas de Processamento Analítico. Por sua vez no terceiro capítulo debruçamo-nos sobre o estudo destes sistemas implementados com base em tecnologia opensource. No quarto capítulo descrevemos o caso de estudo desta dissertação assim como o sistema protótipo criado. No capítulo cinco apresentamos as conclusões e o trabalho futuro. Finalmente apresentamos as referências bibliográficas e os anexos no capítulo seis.



## **2. Sistemas de Informação Geográfica (SIG) e Sistemas de Processamento Analítico (OLAP)**

### **2.1 Introdução**

Desde os tempos mais primitivos que a localização de bens e de matérias-primas é muito importante, independentemente de como estes eram guardados ou transmitidos. Na realidade, nós portugueses, devemos-nos orgulhar dos nossos navegadores que desafiaram as quase inexistentes informações marítimas na altura, mas que seguiram viagem e efectuaram grandes descobertas, detalhando em valiosos mapas, as terras por onde passavam. Esta informação não era só importante para os recursos naturais e alimentos, como também em questões associadas à defesa das pessoas e bens, sendo tipicamente uma tarefa associada às operações militares. Esta informação hoje em dia é tão importante como a de séculos atrás, existindo muita tecnologia somente responsável por avaliar a localização de pessoas e bens.

Em geral, existem várias definições sobre o conceito de Informação Geográfica (IG), sendo na sua maioria baseadas nos mesmos pressupostos ou factos. Segundo

Faria, 2006, a IG corresponde à informação referente a fenómenos ou entidades na superfície terrestre. Esta é então constituída pela localização e pela informação anexa a esta, como por exemplo, “29,9; 9,0; Escola Superior de Tecnologia e Gestão”, sendo que a primeira parte corresponde à localização, a qual deve estar assente num sistema de referência para que a mesma faça sentido e a restante informação corresponde aos atributos dessa localização, onde neste exemplo se apresenta um nome, neste caso “Escola Superior de Tecnologia e Gestão”. Uma característica da IG é que é relativamente estática, ou seja, a informação depois de impressa em papel torna-se estática, não sendo de rápida alteração quer represente um fenómeno natural ou criado pelo homem. Por outro lado, esta informação também se distingue pelo volume de dados necessários para a representação de qualquer assunto (Ferreira, 2006).

Neste contexto, ao longo deste capítulo pretendemos apresentar de uma forma geral os conceitos, terminologias e tecnologias associadas aos Sistemas de Informação Geográfica e aos Sistemas de Processamento Analítico.

## **2.2 Sistemas de Informação Geográfica**

Uma das definições de um Sistema de Informação Geográfico (SIG) é a que corresponde a um conjunto organizado de hardware, software, dados geográficos e pessoas, destinados a eficientemente obter, armazenar, actualizar, manipular, analisar e exibir todas as formas de informação referenciada geograficamente (Fu and Sun, 2010). Outros investigadores (Sutton et al., 2009) apresentam que um SIG pode também ser descrito como um sistema constituído por informação geográfica que consiste em dados geográficos os quais são analisados em sistemas de hardware que armazenam, processam e apresentam os dados, usando um conjunto de aplicações que permitem trabalhar sobre os dados georreferenciados. Efectivamente, fazendo uma pesquisa na literatura, constata-se que existem várias definições para definir um SIG, no entanto, todas as definições se centram nas características associadas à capacidade de recolha, armazenamento, manipulação, análise e visualização de dados georeferenciados e dos seus relacionamentos.

Em termos de características associadas aos SIG é comum a sua caracterização ser estruturada segundo o seu ciclo de vida, pela sua representação de dados, pelo tipo de armazenamento e pelas particularidades das operações de análise espacial. Neste sentido nas secções seguintes apresentamos de uma forma geral uma abordagem a cada uma destas características.



## 2.2.1 Ciclo de Vida

O ciclo de vida de um SIG é estruturado por um conjunto de fases (figura 1), assentes basicamente em cinco tarefas (figura 1): recolha de dados, entrada de dados, análise de dados, informação para a tomada à decisão e a própria tomada à decisão assente na informação dos próprios SIG (ou eventualmente complementada com outro tipo de informação).

Além destas fases principais o ciclo de vida pode compreender outras tarefas ou actividades mais específicas entre cada uma das etapas apresentadas. Por exemplo, entre a fase de recolha e entrada de dados é comum existir uma actividade denominada por “pré-processamento e limpeza de dados” a qual tem como objectivo seleccionar e tratar a informação com mais relevância e importância para a sua utilização e exploração por parte dos agentes de decisão.



Figura 1- Ciclo de vida de um Sistema de Informação Geográfico

Uma outra particularidade a salientar é que, ao longo do tempo, surge a necessidade de se proceder à actualização de informação (tipicamente grandes volumes de dados) no sentido de actualizar os repositórios de dados para que as ferramentas de análise sejam úteis para os agentes de decisão.

## 2.2.2 Representação dos dados

A representação dos dados geográficos tem uma particular importância para permitir o manuseamento da informação, tornando-se necessário assegurar modelos de dados para representar, armazenar e manipular dados georreferenciados. A informação do mundo real é descrita, em geral, de duas maneiras (Lo and Yeung, 2007): como objectos discretos e finitos (como por exemplo, edifícios, estradas e limites de freguesias ou distritos) e como fenómenos com uma distribuição contínua ao longo do espaço geográfico (como a temperatura, a precipitação e o relevo).

Tipicamente, nos SIG, os modelos de dados mais comuns são o modelo vectorial e o modelo matricial (também, conhecido por modelo “raster”). No primeiro caso, este tipo de representação de dados é utilizado para representar objectos geográficos com fronteiras bem definidas, sendo estruturado em pontos (um par de coordenadas X, Y), linhas (dois ou mais pares de coordenadas X,Y) e polígonos (três ou mais pares de coordenadas X,Y, com a primeira e a última coordenada coincidentes). As fotografias áreas podem ser consideradas um tipo de modelo “raster”. Este modelo, corresponde a representações matriciais da informação, ou seja, a informação é dividida em células que por norma tem a forma de um quadrado, mas também podendo tomar a forma de rectângulos, triângulos e hexágonos. Uma particularidade é que a qualidade da informação (resolução da imagem no momento da sua visualização) é directamente proporcional à dimensão das células, ou seja, quanto maior for a célula menor será a resolução e vice-versa (Faria, 2006). Na figura 2 ilustra-se um exemplo de caracterização do tipo de estrutura de dados de um SIG.

Neste tipo de sistemas, a informação geográfica é tipicamente estruturada por camadas ou *layers*, consistindo cada uma num conjunto seleccionado de objectos (discretos e finitos) associados, e respectivos atributos.

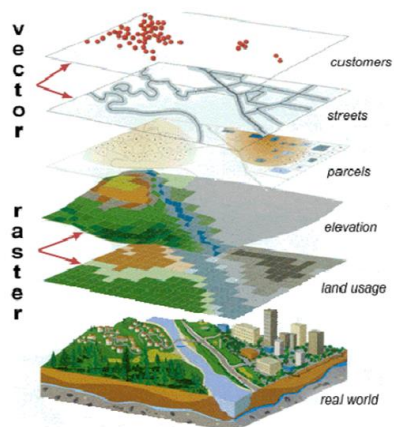


Figura 2- Organização dos dados de um SIG (Fu & Sun, 2010).

Na figura 3 ilustra-se um esquema da exemplificação da caracterização da estrutura em camadas de um SIG, neste caso representando as camadas da realidade do terreno referentes às estradas, dos edifícios e localização dos clientes.

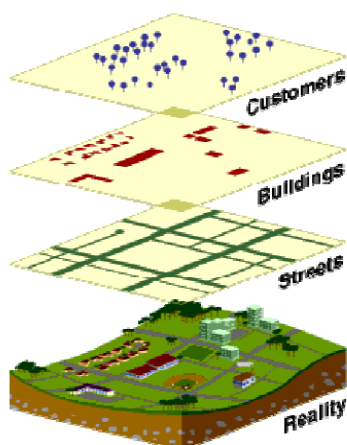


Figura 3- Caracterização da estrutura em Camadas de um SIG (Fu & Son, 2010).

Este tipo de representação de dados é mapeado num sistema de armazenamento, o qual reflecte a semântica e funcionalidades associadas aos actuais sistemas de gestão de bases de dados, assim como às potencialidades de exploração desses mesmos dados, recorrendo à tradicional linguagem de invocação de questões, o SQL – Structured Query Language. Desta forma, este tipo de linguagem é estendido para suportar questões ou pesquisas incorporando os três tipos de dados geométricos vectoriais, nomeadamente, os pontos, as linhas e os polígonos.

Na figura 4 ilustra-se um exemplo da representação num sistema de gestão de base de dados associados a um SIG. Em geral, os sistemas de gestão de bases de dados apresentam módulos de extensibilidade para tratar a componente georreferenciada dos dados, não sendo sistemas autónomos fora dos sistemas de gestão de bases de dados (Steiniger & Hunter, 2010). Nesta ilustração evidencia-se que o polígono do mapa (“Geleira Ecology”) tem um identificador (“71”) é armazenado como um tipo de dados polígono, o qual internamente armazena um conjunto de pontos referente a esse polígono.

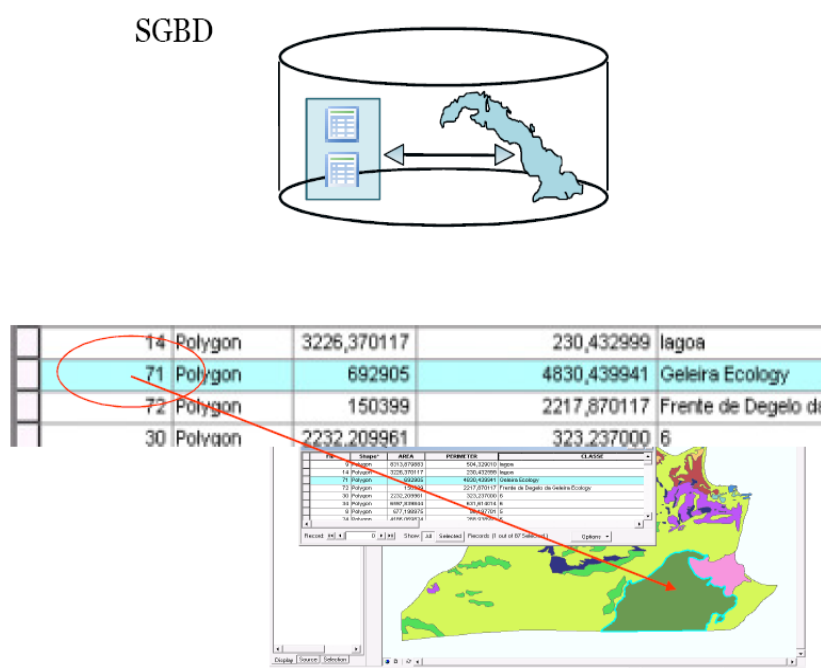


Figura 4- Representação num Sistema de Gestão de base de dados dos dados associados a um SIG.

### 2.2.3 Armazenamento de Informação Geográfica

Ao longo dos anos, em virtude da utilização massiva deste tipo de sistemas, o volume de crescimento das bases de dados tem sido elevado, muito devido aos melhoramentos tecnológicos e à facilidade de utilização de vários equipamentos electrónicos com a capacidade de recolher dados, como, por exemplo, os telemóveis com GPS, automóveis com sistema de tracking, sistemas de monitorização ambientais, entre outros. Considerando o vasto tipo de dados, a informação pode ser armazenada de várias formas, como por exemplo, em imagens, em ficheiros de texto, tanto para o

modelo vectorial como para o modelo *raster*. No que se referenre ao presente projecto, o modelo que será utilizado é o modelo vectorial, com o qual é possível representar a realidade através de três elementos: pontos, linhas e polígonos. Para armazenar este tipo de informação seguindo a representação vectorial é possível se utilizar diversos formatos, como um sistema de ficheiros ou através de bases de dados.

Em termos de armazenamento de informação geográfica em ficheiros no modelo vectorial, o tipo mais comum é o designado por *Shapefiles*, o qual foi criado pela ESRI<sup>1</sup>, nos finais de 1997 para uma das suas aplicações, o *ArcView*<sup>2</sup>, sendo constituído por um conjunto de ficheiros com o mesmo nome de base e diferentes extensões (“*.shp*”, “*.dbf*”, entre outros). Contudo, há a salientar que, apesar de ser criado por uma empresa privada, actualmente este formato é livre. Em termos de estrutura, este ficheiro armazena localizações geométricas e informação de atributos associada a estes num formato vectorial. A informação da geometria de cada entidade está associada a um conjunto de coordenadas com valores reais. Estas geometrias podem ser simples como um ponto ou uma linha, mas podem também evoluir para polígonos. Cada entidade tem um conjunto de atributos que estão armazenados no formato dBASE (Faria, 2006, ESRI, July 1998). Neste tipo de armazenamento para se construir uma shapefile são necessários no mínimo três ficheiros:

- **.shp** – armazena as entidades geométricas;
- **.shx** – armazena os índices das entidades geométricas;
- **.dbf** – ficheiro dBASE que armazena os atributos das entidades.

No entanto, apesar dos três ficheiros acima descritos serem os necessários para construir uma shapefile, podem ser adicionados mais ficheiros para que o desempenho das análises e visualizações seja melhorada, ou mesmo para guardar metadados, podendo o conjunto de ficheiros ser alargado para o seguinte tipo de extensão e respectivo conteúdo:

- **.sbn** ou **.sbx** – armazena o índice espacial das entidades;
- **.fbn** ou **.fbx** – armazena o índice espacial das entidades (apenas para *shapefiles* de leitura);
- **.ain** ou **.aih** – armazena os índice do atributo dos campos activos de um tabela;
- **.prj** – Armazena a Informação sobre o sistema de coordenadas;

---

<sup>1</sup> Enviromental System Ressearch Institute [www.esri.com](http://www.esri.com)

<sup>2</sup> <http://www.esri.com/software/arcview/>

- **.shp.xml** – armazena os metadados.

Para se proceder à visualização, análise e exploração deste tipo de ficheiros, existem muitas aplicações quer proprietárias quer baseadas em software livre como por exemplo, as ferramentas gvSIG<sup>3</sup>, uDig<sup>4</sup>, OpenJump<sup>5</sup>, Quantum GIS<sup>6</sup>, entre outros.

Em termos de armazenamento da informação em bases de dados e, considerando que com volumes de dados cada vez maiores, o armazenamento em sistemas de ficheiros não é viável. Por isso, os Sistemas de Gestão de Base de Dados (SGBD) evoluíram para que se pudesse armazenar este tipo de informação. Neste tipo de armazenamento também é possível escolher *software* proprietário ou open source. Para soluções com custos para o utilizador, pode-se evidenciar, por exemplo, os seguintes Sistemas de Gestão de Base de Dados: Microsoft SQL Server e a Oracle e, para soluções opensource: o PostgreSQL + PostGIS e o MySQL. Neste trabalho utilizamos o PostgreSQL, dado o seu carácter livre, uma vez que permite a integração da ferramenta PostGIS, a qual corresponde a um conjunto de funções para que seja possível a manipulação dos dados geográficos. Neste sentido, iremos de seguida basear a descrição do nosso trabalho seguindo a estrutura interna do PostGIS.

Em termos de análise do PostgreSQL para armazenamento de informação geográfica o módulo PostGIS tem de ser instalado no servidor após a instalação da versão base do PostgreSQL. Neste sistema de gestão de bases de dados, ao criar uma base de dados com suporte à componente geográfica, são geradas duas tabelas no *schema public*, de acordo com a especificação *Simple Feature for SQL do Open GIS Consortium*<sup>7</sup>. As tabelas referidas têm a designação de “Spatial\_ref\_sys” e de “Geometry\_columns”. A tabela “Spatial\_ref\_sys” armazena todos os Sistemas de Coordenadas, conhecidos por SRID (Spatial Reference System Identifier). Estes sistemas são definidos pelo EPSG - European Petroleum Survey Group<sup>8</sup>.

A figura 5 ilustra os metadados de um Sistema de Coordenadas (neste caso o 27493). Em termos de sistemas de coordenadas mais utilizadas em Portugal de acordo com Instituto Geográfico do Exército (IGEOE<sup>9</sup>) são os seguintes:

---

<sup>3</sup> <http://www.gvsig.org/web/>

<sup>4</sup> <http://udig.refractions.net/>

<sup>5</sup> <http://www.openjump.org/>

<sup>6</sup> <http://www.qgis.org/>

<sup>7</sup> <http://www.opengeospatial.org/standards/sfa>

<sup>8</sup> <http://www.epsg.org/>

<sup>9</sup> [http://mapas.igeo.pt/igp/epsg\\_codes.html](http://mapas.igeo.pt/igp/epsg_codes.html)

- EPSG: 27492 (Datum 73/ Hayford-Gauss)
- EPSG: 20790 (Datum Lisboa/ Coordenadas Militares)
- EPSG: 20791 (Datum Lisboa/ Hayford-Gauss)
- EPSG: 4326 (WGS 84/ Coordenadas Geográficas)
- EPSG: 4258 (ETRS89/ Coordenadas Geográficas)
- EPSG: 2191 (Datum Base SE - Porto Santo (Madeira) / UTM zona 28N)
- EPSG: 2188 (Datum Observatório - Flores (Grupo Ocidental do Arquipélago dos Açores) / UTM zona 25N)
- EPSG: 2189 /Datum Base SW - Graciosa (Grupo Central do Arquipélago dos Açores) / UTM zona 26N)
- EPSG: 2190 (Datum S. Braz - S. Miguel (Grupo Oriental do Arquipélago dos Açores) / UTM zona 26N)
- EPSG: 3763 (PT-TM06/ETRS89)

spatial_ref_sys	
<b>srid</b>	27493
<b>auth_name</b>	EPSG
<b>auth_srid</b>	27492
<b>srtext</b>	PROJCS["Datum 73 / Modified Portuguese Grid(deprecated)", GEOGCS["Datum 73", DATUM["Datum_73", SPHEROID["International 1924", 6378388, 297, AUTHORITY["EPSG", "7022"]], AUTHORITY["EPSG", "6274"]], PRIMEM["Greenwich", 0, AUTHORITY["EPSG", "8901"]], UNIT["degree", 0.01745329251994328, AUTHORITY["EPSG", "9122"]], AUTHORITY["EPSG", "4274"]], UNIT["metre", 1, AUTHORITY["EPSG", "9001"]], PROJECTION["Transverse_Mercator"], PARAMETER["latitude_of_origin", 39.66666666666666], PARAMETER["central_meridian", -8.131906111111112], PARAMETER["scale_factor", 1], PARAMETER["false_easting", 180.598], PARAMETER["false_northing", -86.99], AUTHORITY["EPSG", "27492"], AXIS["Y", EAST], AXIS["X", NORTH]]
<b>proj4text</b>	+proj=tmerc +lat_0=39.66666666666666 +lon_0=-8.131906111111112 +k=1 +x_0=180.598 +y_0=-86.98999999999999 +ellps=intl +units=m +no_defs

Figura 5 – Ilustração do conteúdo da tabela “spatial\_ref\_sys”.

A segunda tabela de nome “Geometry\_columns” (figura 6) deve ser preenchida com todas as colunas de todas as tabelas que sejam do tipo geométrico, ou seja, corresponde à tabela de metadados das colunas com geometria. Por exemplo, na figura 6 ilustra-se um exemplo do registo das colunas geométricas designadas por “the\_geom” para as tabelas “fregvalimar” e “limitept”.

oid	f_table_catalog	f_table_schema	f_table_name	f_geometry_name	coord_dimension	srid	type
21258	*	public	fregvalmar	the_geom	2	-1	MULTIPOLYGON
21450	*	public	limept	the_geom	2	-1	MULTIPOLYGON

Figura 6 – Ilustração dos registos da tabela “Geometry columns”.

O objectivo destes registos é permitir ao módulo suportar informação geográfica e permitir o processamento dos pedidos de visualização oriundos de ferramentas de visualização como, por exemplo, o Quantum Gis<sup>10</sup> ou através da invocação de comandos SQL com funções aplicadas à componente geográfica.

### 2.3 Sistemas de Processamento Analítico

A informação corresponde a um dos bens mais valiosos de uma organização, reflectindo-se nela anos e anos do seu trabalho. A análise das várias vertentes de negócio sobre a informação acumulada, veio ajudar os gestores a tornarem as suas decisões mais sustentadas e consistentes. Historicamente, os sistemas de informação desenvolvidos, tentam analisar grandes volumes de informação acumulada nos sistemas operacionais, que na maioria das vezes se encontram em fontes heterogéneas. No entanto, a sua capacidade para responder a análises complexas vai diminuindo com o aumento do volume dos dados, deixando escapar potenciais análises que por vezes são chaves de sucesso para o futuro de uma organização. Por sua vez, os Sistemas de Data Warehousing (Berson & Smith 1997) permitem fazer a conjugação das várias fontes de informação integrando-as num único repositório de informação com a particularidade de ser orientado por áreas de negócio, estando os seus registos com uma referência temporal, permitindo aos gestores analisarem todo o histórico dos dados ao longo do tempo.

Neste contexto, esta secção apresenta de uma forma clara e sucinta os desafios na implementação de um Sistema de Data Warehousing, analisando o porquê da necessidade do seu planeamento, quando e como se implementa, assim como uma descrição das características e particularidades associadas aos sistemas de processamento analítico.

<sup>10</sup> <http://www.qgis.org/>



### **2.3.1 Sistemas de Data Warehousing**

Um Data Warehouse (DW) (Inmon, 1996) (Kimball, et al. 1998) é vulgarmente conhecido como um repositório de informação com características específicas, onde a informação está estruturada e organizada de forma a permitir que a análise por parte dos gestores das organizações seja a pretendida e disponível no menor espaço de tempo possível, assim como assegurar um elevado grau de qualidade e fiabilidade, no sentido a auxiliá-los nas decisões que irão tomar. De modo a facilitar a análise, a informação pretendida deverá estar armazenada num repositório central de uma forma bem definida e estruturada, de maneira a conjugar toda a informação, organizando-a por áreas de negócio, assim como manter o histórico do seu conteúdo em termos de tempo, permitindo responder o mais eficientemente possível às “questões” de análise invocadas. Este é o objectivo de um Data Warehouse, o qual é caracterizado por quatro características essenciais (Inmon, 1996):

- **Organizada por assuntos:** A informação está organizada por assuntos/temas referentes à área de negócio ou de actividade em que a organização está inserida e que os gestores pretendem analisar. Ou seja, o contexto dos temas da informação a analisar é perfeitamente claro e definido, correspondendo às necessidades dos agentes de decisão.
- **Informação Integrada:** a informação está armazenada num repositório central. Nos sistemas operacionais a informação pode estar repartida por fontes heterogéneas e dispersas.
- **A informação é não volátil:** num sistema de Data Warehouse a informação existente não é alterada, a partir do momento em que se inserem blocos de informação extraídos dos sistemas operacionais, apenas se pode ler informação, nunca alterar, permitindo que a informação seja consistente.
- **Informação Temporal:** a informação possui uma referência temporal. Esta referência permite manter o histórico da informação, podendo ser analisada até à granularidade (em termos de tempo tal como: dia, mês, semestre, trimestre, ano, feriados, pontes, entre outros), ou seja, o nível mais baixo a que estão os dados.

A tabela 1 (Bédard et al, 2001) apresenta a diferença entre os modelos relacionais e os multidimensionais, onde se evidencia o nível de agregação, o nível de optimização, o tipo de dados e o tipo de estrutura.

<b>Modelos Relacionais (ex. ERP)</b>	<b>Modelos Multidimensionais (ex. DW)</b>
Optimizado para transacções	Optimizado para análises, decisões
Origem dos dados	Cópia de dados
Dados detalhados	Dados agregados
Orientados a aplicação	Orientado à Organização
Dados diários/correntes	Dados correntes e histórico
Estrutura Normalizada	Desnormalizado, Redundante

Tabela 1 – Diferenças entre Sistemas Relacionais e Multidimensionais (Bédard et al, 2001)

Em geral, um Data Warehouse armazena amostras (ou seja os anos de histórico que a organização considere que são relevantes para o seu negócio) de informação dos sistemas operacionais. Por este facto, o Data Warehouse tem todos os estados da informação, menos o actual. Por outro lado, os dados nos sistemas operacionais estão (ou devem) estar normalizados por forma a manter a integridade da informação, enquanto num Data Warehouse a informação pode não estar normalizada, o que permite melhorar o desempenho das análises invocadas pelos gestores através de ferramentas de acesso aos dados.

Pode-se então dizer que um Data Warehouse não é mais que um conjunto de vários Data Marts que se podem relacionar entre si, ou serem complemente individuais. Contudo como se pode verificar na tabela 2, existem algumas diferenças entre estas duas entidades.

<b>Data Warehouse</b>	<b>Data Mart</b>
Construído para análises	Construído para análises de alto nível
Agregação de dados	Alto nível de agregação de dados
Orientado à organização	Orientado ao negócio
Desnormalizado, redundante	Altamente desnormalizado, redundante

Tabela 2 – Diferenças entre Data Warehouse e Data Mart

Por outro lado, é com base na estrutura dos Data Marts do DW que são geradas todas as combinações de agrupamento sendo materializado nos tradicionais Cubos de dados ou Estruturas Multidimensionais.

### **2.3.2.1 Porquê da necessidade de um Data Warehouse**

Considerando a necessidade das organizações obterem informação de qualidade o mais rápido possível de modo a auxiliar os seus gestores a tomarem decisões, assim como a análise de grandes volumes de dados segundo várias vertentes de negócio, a necessidade de desenho e implementação de um sistema de suporte a informação analítica é justificada pelos seguintes cenários:

- Quando o problema se centra no desempenho na análise dos sistemas de suporte à decisão;
- Execução de tarefas associadas com a resposta a questões/pesquisas complexas e na elaboração de relatórios em servidores não suportados pelos sistemas transacionais;
- Fornecimento de um ambiente no qual seja necessário pouco conhecimento sobre as bases de dados existentes para a elaboração e manutenção de pesquisas e de relatórios por parte dos utilizadores;
- Utilização de modelos de dados e tecnologias orientadas à resolução de pesquisas e à elaboração de relatórios com dados agrupados;
- Fornecimento de um repositório de dados tratados que possa ser usado para responder a pesquisas analíticas ou para a elaboração de relatórios;
- Facilitar a resposta a pesquisas e a elaboração de relatórios sobre dados provenientes de diversos sistemas e até de dados em sistemas de armazenamento secundários;
- Prevenção no acesso aos sistemas transaccionais de utilizadores que apenas necessitem de respostas a pesquisas, bem como elaborar relatórios sobre assuntos relacionados com actividades operacionais da organização (ou de outros dados externos).

### **2.3.2.2 Planeamento de um Data Warehouse**

A decisão da implementação de um Data Warehouse é uma ponderação importante para uma organização, uma vez que para que esta implementação seja bem-sucedida tem de se envolver toda a organização de modo que todos os colaboradores que a vão utilizar tenham confiança nos dados que vão analisar. O ciclo de vida típico de um projecto de Suporte à Decisão (figura 7), é caracterizado na generalidade dos

casos, por ser organizado em quatro fases (Kimball et al. 1998) nomeadamente: a fase da definição de requisitos, a fase da análise e desenho-planeamento, a fase do desenvolvimento e a fase da instalação e formação.

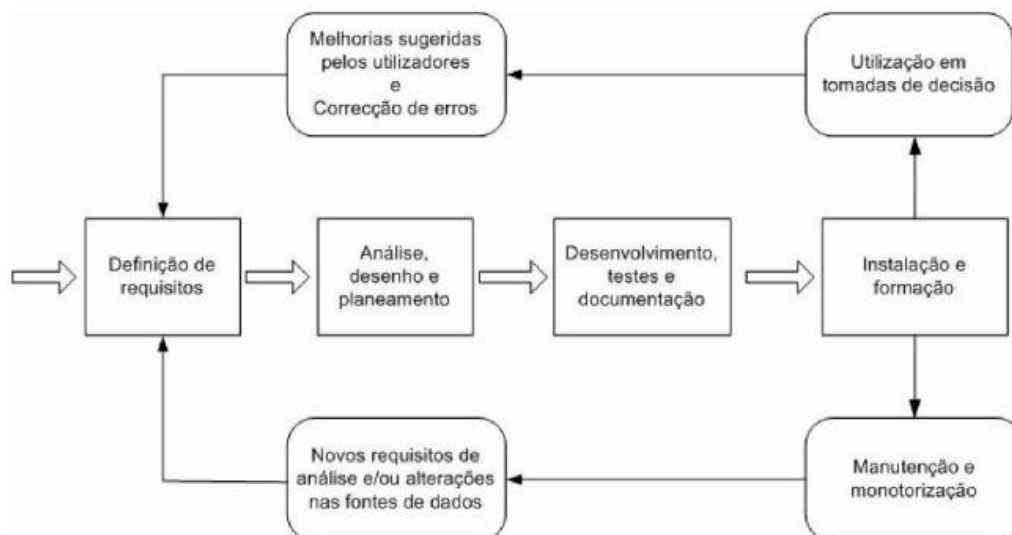


Figura 7 - Ciclo de vida típico de um projeto de Suporte à Decisão (Microsoft)

Cada uma destas fases é constituída por um conjunto de tarefas que devem ser desempenhadas consoante o processo que esteja a ser efectuado. A fase da definição de requisitos centra-se no estudo da necessidade de implementação de um DW em particular no levantamento de requisitos para a elaboração do projecto numa determinada área de negócio. A fase de desenho corresponde à elaboração do desenho lógico da base de dados de suporte à informação das várias fontes de dados de modo a ser orientada para o objectivo de negócio a que o repositório analítico (DW) se destina. Também nesta fase são definidos todos os processos de tratamento das tarefas de selecção, extracção, transformação e integração dos dados no DW, bem como no desenho das aplicações com que os utilizadores irão analisar os dados. A fase de desenvolvimento centra-se na implementação das tarefas e processos especificados na fase de desenho assim como efectuar testes a todo o sistema, bem como documentar todos os processos implementados. Por fim, na fase de instalação/formação, é instalada a solução final, acompanhada de formação aos utilizadores, no sentido de tirarem partido de todas as potencialidades do sistema.

### **2.3.3.Características dos Sistemas de Processamento Analítico**

Os agentes de decisão das organizações sentem cada vez mais a necessidade de analisarem a informação oriunda de várias fontes assim como da informação histórica armazenada nos sistemas operacionais das suas organizações. Pretendem analisar grandes volumes de dados, segundo várias vertentes de negócio, eventualmente complementados com outra informação externa, mas pretendem aceder a essa mesma informação de uma forma rápida e expedita. Contudo, este tipo de análises são caracterizadas por envolver várias variáveis complexas, grandes volumes de dados, e conjugados com várias dimensões de tempo e de negócio. Conforme já referimos, os sistemas operacionais estão desenhados para suportarem informação associada às actividades operacionais das organizações, não estando preparadas para suportar ou responder de um modo rápido a pesquisas ou à obtenção de informação analítica e agregada requerida pelos agentes de decisão, os quais são os principais intervenientes na análise de dados.

Através de uma modelação dimensional dos dados, os Sistemas de Processamento Analítico (Berson & Smith, 1997) permitem apresentar os dados numa arquitectura simples e intuitiva, contendo toda a informação pretendida pelos agentes de decisão, permitindo às ferramentas de suporte à decisão trabalharem sobre os dados armazenados nessas estruturas, de uma forma rápida, de modo a rentabilizar o tempo de resposta às análises pretendidas por esses agentes. Contudo, neste tipo de sistemas OLAP existem várias particularidades que devem ser tidas em conta no seu processo de desenho e implementação de modo a serem úteis para os fins a que se destinam, nomeadamente na disponibilização da informação analítica que os agentes de decisão necessitam para que estes usem ferramentas simples para explorarem os grandes volumes de dados armazenados nos sistemas de Data Warehousing. Neste sentido, no processo de desenho e implementação deste tipo de sistemas tem de ser tidas em consideração as particularidades associadas às seguintes temáticas:

#### **a) Modelação de Dados**

A modelação de dados definidas nos repositórios de dados analíticos ou sistemas de Data Warehouse além de terem de assegurar as características apresentadas na secção 2.3.2, nomeadamente a informação estar organizada temporalmente e por assunto, a definição do desenho da sua estrutura tem de seguir o conceito de

modelação dimensional. A modelação dimensional dos dados corresponde a uma técnica de desenho lógico que constrói uma estrutura de armazenamento de informação de fácil entendimento e com objectivo de melhorar o desempenho no tempo de resposta de acesso aos dados.

O resultado deste tipo de modelação é uma estrutura (ou tabela de factos) em “estrela” ou “flocos de neve” sendo caracterizado por uma tabela principal (onde são definidas as chaves de dimensão e as medidas) e por um conjunto de tabelas ligadas directamente de modo a rentabilizar o desempenho das operações de junção, as quais degradam o desempenho de pesquisas quando se envolve várias variáveis de negócio (ou dimensões). O objectivo destas tabelas de dimensão é complementar a informação por parte dos agentes de decisão ou processos de análise.

As figuras 8 e 9 exemplificam de forma muito genérica os tradicionais modelos em “estrela” e em “flocos de neve”. O “modelo em estrela” caracteriza-se por uma tabela de factos rodeada de várias tabelas de dimensão, não estando estas normalizadas. Por sua vez o modelo em “flocos de neve” corresponde a uma tabela de factos rodeada de tabelas de dimensão ligadas a um conjunto de tabelas normalizadas as quais permitem sub categorizar os atributos de uma tabela de dimensão.

No âmbito da modelação dimensional (Inmon, 1996) (Kimball, et al. 1998) são apresentados e definidos os seguintes termos: dimensões, atributo, medida, hierarquia, tabela de factos, granularidade, data mart e estrutura multidimensional ou cubo de dados. O conceito de *Dimensões* é utilizado para catálogos, que por sua vez, são os dados mais relevantes para a organização. Entre os mais importantes estão sempre dados como: produtos, lojas, clientes, ente outros. As dimensões estão ainda relacionados a atributos e a hierarquias. Os atributos são dados como por exemplo nome do cliente/produto, morada, isto é algo que não seja possível agregar. Numa dimensão também é possível criar hierarquias, como uma estrutura mercadológica<sup>11</sup> da organização.

---

<sup>11</sup> Mercadológica: Por exemplo a estrutura de produtos dentro de um supermercado (Companhia -> Departamento -> Sub-Departamento-> Produto)

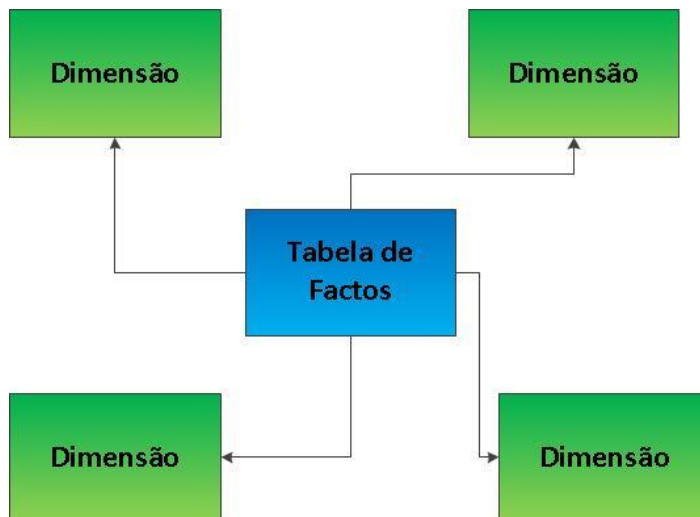


Figura 8 - Esquemático tradicional de um Data Mart de um Data Warehouse usando uma estrutura em estrela.

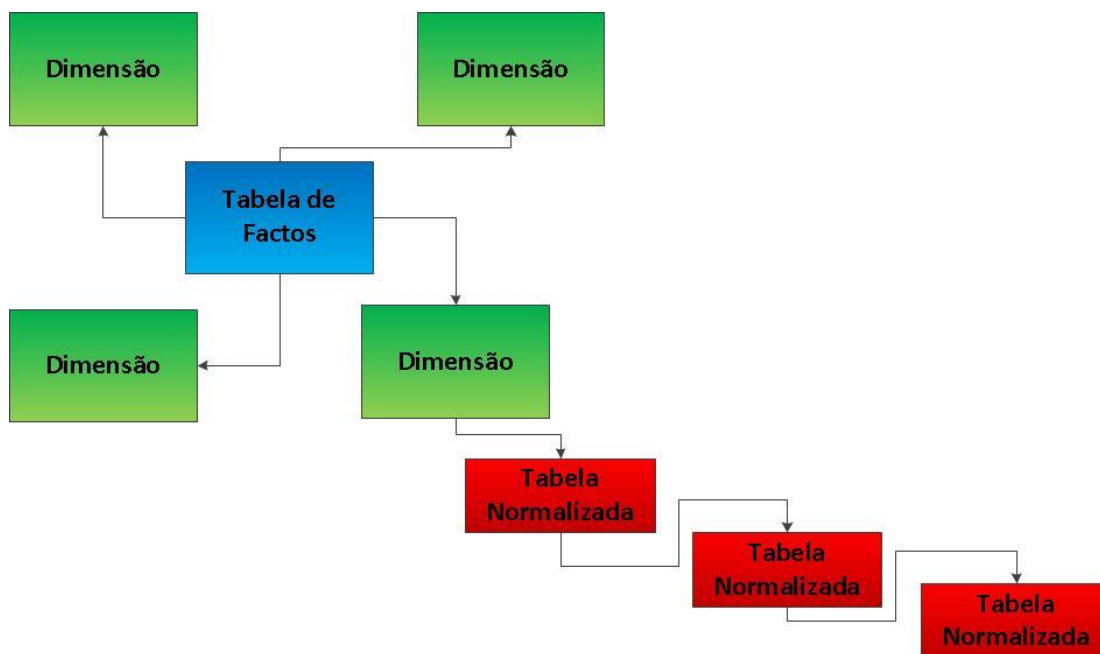


Figura 9 - Esquemático tradicional de um Data Mart de um Data Warehouse usando uma estrutura em floco de neve.

**b) Processo de Selecção, Extracção, Transformação e Carregamento.**

Conforme referido anteriormente, as ferramentas de processamento analítico processam previamente grande parte dos cálculos pretendidos pelos agentes de

decisão de modo a que utilizem ferramentas de análise de dados de uma forma rápida e intuitiva. Por sua vez, nestes dados é comum a existência de algumas dificuldades no sentido de assegurar a coerência de informação, como por exemplo: a necessidade de efectuar conversões de tipos de dados, o tratamento de valores nulos nos dados dos sistemas operacionais, a existência de informação em duplicado, o uso incorrecto de códigos de caracteres, a necessidade de complementar a informação existente nas dimensões do DW, entre outros problemas communmente encontrados nos processos de importação. Neste sentido, é usual a utilização de processos orientados para a selecção, extracção, limpeza, transformação e integração de dados no DW, podendo por vezes existir uma zona temporária de concentração de dados entre os Sistemas Operacionais e os Sistemas de DW. Na figura 10 ilustra-se uma esquemática geral sobre o processamento de estruturas multidimensionais de dados em que o processo se inicia pelas tarefas de *Extract Transform Load* (ETL) sobre dados dos sistemas operacionais até ao seu armazenamento no DW. Após um modelo definido torna-se necessário criar os processos que permitem que o modelo receba dados de diversas fontes. Este processo deve ser bem pensado pela organização, de modo a integrar todos os dados que sejam relevantes, e até pensar num futuro sistema que forneça esses dados. Com este tipo de análise e organização deve-se escolher a ferramenta que melhor se adequa a situação.

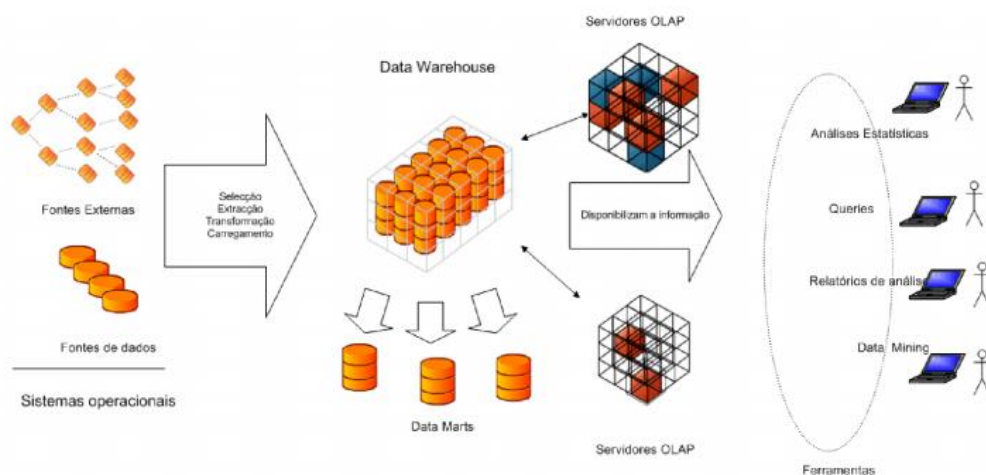


Figura 10- Processamento de Estruturas Multidimensionais de Dados.

Esta fase não pode ser somente encarada como a transformação de dados de um tipo para outro, tendo aqui particular atenção aos dados geográficos/espaciais. Durante



o processo podem ocorrer muitas situações complexas tal como a reestruturação de objectos de forma a obter um *merge* da informação que se encontra em determinados sistemas, se não estão relacionados entre si.

A figura 11 ilustra um o processo ETL. Tipicamente este processo centra-se nas seguintes tarefas:

- **Seleção e tratamento dos dados:** tem como objetivo seleccionar os dados dos vários repositórios de dados (sistemas operacionais, Data Warehouses). Nesta selecção e tratamento dos dados tenta-se eliminar o espaço de pesquisa, eliminando atributos que não têm qualquer interesse no processo de descoberta de conhecimento. Em suma, a tarefa do tratamento dos dados centra-se na limpeza e verificação dos dados, como por exemplo, na desduplicação de registos e a verificação de inconsistências de informação.
- **Transformação:** tem o objetivo de tornar a pesquisa de informação mais eficiente através da agregação de dados. Saliencia-se que a forma como os dados são codificados ou agregados influencia fortemente os resultados que poderão ser obtidos na fase de Interpretação. Nesta fase, é comum, por exemplo efectuar o agrupamento dos dados através de hierarquias de conceitos e de classes de valores que permitam a execução destas tarefas.

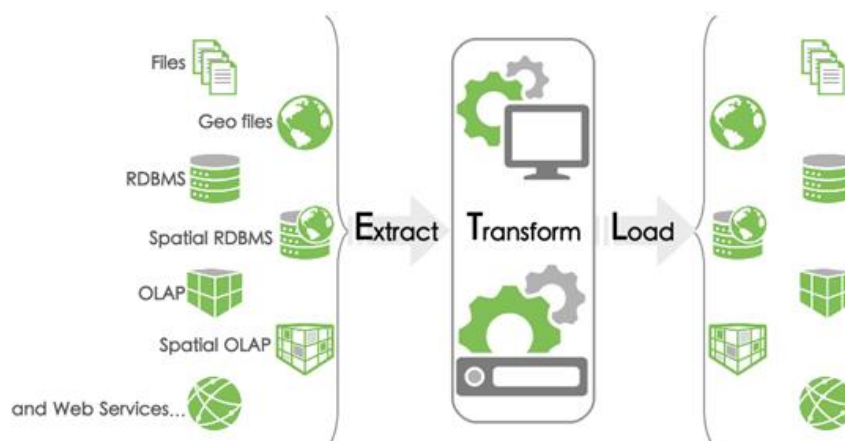


Figura 11 - Ilustração do processo de Extracção, Transformação e Carregamento  
fonte: (<http://www.spatialytics.org/projects/geokettle/>)

### c) Processamento de Estruturas Multidimensionais de Dados

Considerando o volume de dados, o número de dimensões e o número de medidas na estrutura dimensional definidos no DW, o processamento destas estruturas executa-se procedendo à computação da maioria das combinações dos agrupamentos (“operação de “Group-by”) sobre todas as dimensões dos Data Marts que constituem o DW. Neste sentido, este processamento é uma tarefa computacional bastante difícil requerendo bastantes recursos de memória, processador e tempo, além de algum custo com espaço para armazenar o resultado deste processamento.

O resultado do processamento das combinações dos agrupamentos sobre todas as dimensões dos Data Marts, são denominadas por Cubos de Dados (Gray et al, 1996). Na figura 12 apresenta-se um exemplo de um cubo de dados, o qual corresponde ao modelo multidimensional que se implementa sobre as tabelas de factos e dimensões dos Data Marts, podendo cada cubo utilizar apenas uma ou várias tabelas de factos.

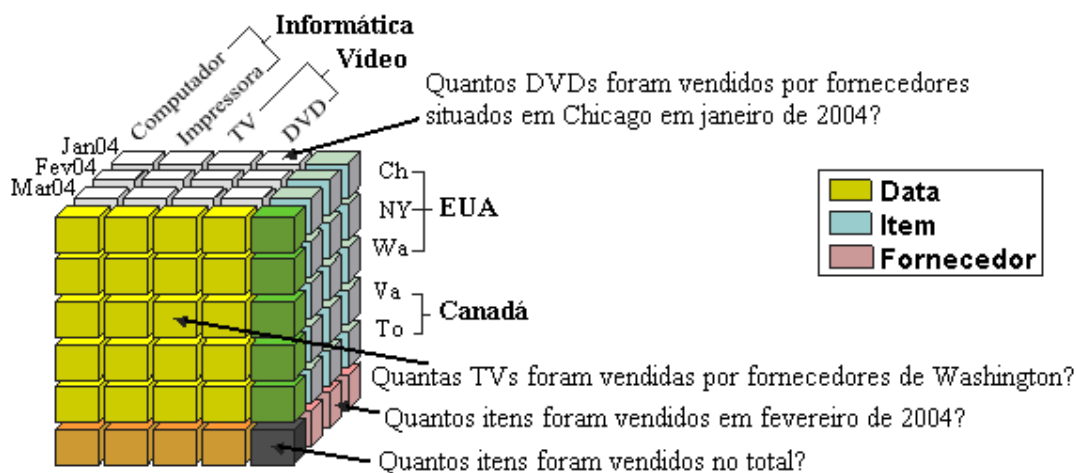


Figura 12 – Ilustração da estrutura geral de visualização de um cubo de dados (Siqueira, 2009)

### d) Operações de Análise de Dados Analíticos

O OLAP é conhecido como uma arquitectura de base de dados a qual é estruturada e desenhada de maneira a permitir às ferramentas de apoio à decisão um manuseamento fácil sobre os dados permitindo aos utilizadores

“navegarem” sobre a informação conjugando várias dimensões de análise, assim como permitir aos agentes de decisão ou às ferramentas de análise de dados o acesso a pesquisas de análise complexas obtendo o menor tempo de resposta possível. Tal como referimos, isto é conseguido devido à utilização da modelação dimensional dos dados a qual reflecte uma orientação para o negócio definido pelos agentes de decisão. Neste sentido, as operações disponibilizadas por estes sistemas OLAP permitem:

- a) Análise de informação com base em hierarquias;
- b) Visualização através de vistas de dados, de subconjuntos de informação (vulgarmente designadas de operações de “Slicing”);
- c) A análise de dados agregados (“Roll-up” e “Drill-down”)
- d) A análise de conjugação de várias vistas em termos de comparação (“Rotation”).

Tal como num DW “tradicional” existem as operações de Roll-UP, Drill-Down, Slice, Dice, num sistema data warehousing com dados geográficos estas operações também existem, no mesmo enquadramento lógico, contudo agora é possível a utilização das funções de agregação para os dados espaciais como para os dados alfanuméricos. A existência de dados espaciais não impede em nada a utilização destas funcionalidades sobre os restantes dados. A figura 13 ilustra o resultado destas operações em formato de tabela.

		Measures		
Promotions	Product	Unit Sales	Store Cost	Store Sales
+All Promotions	-All Products	266.773	225.627,23	565.238,13
	+Drink	24.597	19.477,23	48.836,21
	+Food	191.940	163.270,72	409.035,59
	+Non-Consumable	50.236	42.879,28	107.366,33

		Measures		
Promotions	Product	Unit Sales	Store Cost	Store Sales
+All Promotions	-All Products	266.773	225.627,23	565.238,13
	+Drink	24.597	19.477,23	48.836,21
	+Food	191.940	163.270,72	409.035,59
	-Non-Consumable	50.236	42.879,28	107.366,33
	+Carousel	841	595,97	1.500,11
	+Checkout	1.779	1.525,04	3.767,71
	+Health and Hygiene	16.284	12.972,99	32.571,86
	+Household	27.038	24.170,73	60.469,89
+Periodicals	4.294	3.614,55	9.056,76	

Figura 13 – Exemplificação do resultado de operações OLAP usando dados alfanuméricos. Fonte: Exemplo GeoFoodMart do GeoMondrian

**e) Vistas de Dados**

Conforme já referimos neste documento os DW armazenam tipicamente grandes volumes de dados tendo sido desenhados para se obterem respostas rápidas envolvendo várias variáveis de análise nas pesquisas. Uma das estratégias para minimizar o tempo de resposta na obtenção de informação neste tipo de condição é recorrendo à criação de vistas de dados (Gupta & Mumick, 1999) podendo ser definidas como um subconjunto de informação das tabelas de facto das Data Marts. Utilizando este tipo de materialização de dados, consegue-se obter informação rápida, contudo tem de se ter em consideração o nível de qualidade/quantidade da informação que se disponibiliza aos agentes de decisão, assim como a problemática associada à manutenção incremental dos dados no DW e a necessidade da ser reprocessada para que os dados sejam actualizados.

**f) Manutenção incremental**

Tipicamente a manutenção incremental de informação corresponde à tarefa de proceder à actualização dos repositórios analíticos quando há uma actualização de informação nos sistemas operacionais. Trata-se efetivamente de uma tarefa complexa, requerendo metodologias e técnicas específicas para o efeito, como por exemplo a utilização junto das tabelas dos sistemas operacionais de tabelas “fantasma” com triggers para registar as operações de inserção, actualização e remoção de modo a se analisar que alterações foram feitas nas fontes de dados e, assim, proceder á actualização do DW com a nova informação. Apesar de termos noção da problemática desta tarefa, neste trabalho não analisamos com detalhe o processo de actualização futura de dados, mas sim, procedemos ao primeiro carregamento do DW e conseqüente processamento das estruturas multidimensionais, sendo um dos trabalhos futuros a análise deste tipo de tarefas quando aplicados a de informação geográfica.

**g) Tipos de Armazenamento**

Nos tradicionais sistemas de armazenamento de informação analítica é tipicamente comum o armazenamento de dados em sistemas *Relational OnLine Analytical Processing (ROLAP)*, *Multidimensional OnLine Analytical*

*Processing* (MOLAP) e *Hybrid OnLine Analytical Processing* (HOLAP) (Berson & Smith 1997). O esquema de armazenamento ROLAP assenta num sistema de gestão de bases de dados relacional, no qual a informação está assente na mesma infraestrutura que a dos repositórios analíticos. O esquema MOLAP centra-se em lidar com queries como o desempenho na agregação e sumariação de resultados através das operações (agregação e refinamento) mais comuns nestes ambientes. Este tipo de armazenamento de EMD (estruturas multidimensionais de dados) tem a particularidade de colocar todos os dados requeridos pelos serviços OLAP numa base de dados multidimensional especial, usando hierarquias dos dados organizadas em dimensões e medidas por tempo. Por sua vez, no sistema HOLAP são colocadas numa base de dados multidimensional as agregações pretendidas para as queries, de maneira a processar previamente todas essas agregações. Neste trabalho, não nos debruçamos sobre a análise e eficiência da aplicabilidade deste tipo de sistemas. O sistema base de suporte ao projecto seguiu a orientação ROLAP, sendo no futuro um dos trabalhos de análise a possível comparação de desempenho dos sistemas quando aplicados a grandes volumes de dados contendo informação geográfica.

## **2.4 Reflexão Crítica**

Neste capítulo apresentamos, de uma forma geral, os conceitos associados aos Sistemas de Informação Geográfica e aos Sistemas de Processamento Analítico. Podemos constatar que ambos os sistemas requerem o estudo e assimilação de um conjunto alargado de conceitos e terminologias, como entender a sua forma de armazenamento e de processamento para disponibilizar aos agentes de decisão um conjunto de funcionalidades de análise de dados (materializada através de ferramentas de visualização de dados como as ferramentas de Business Intelligence), assim como a disponibilização de dados agregados e tratados para se aplicarem técnicas associadas à Descoberta de Conhecimento em Base de Dados.

No que se refere aos conceitos teóricos (Inmon, 1996) (Kimball, et al. 1998) associados aos SDW e às ferramentas OLAP (Berson & Smith 1997), os mesmos tem sido consolidados desde a década de 90, tendo sido aceite com sucesso a sua viabilidade e utilidade aplicada aos tradicionais dados alfanuméricos. Com a

consolidação das vantagens dos SDW, das funcionalidades das ferramentas OLAP e das boas práticas para o planeamento e implementação deste tipo de sistemas, consegue-se aliar às vantagens da informação dos Sistemas de Informação Geográfica, quer pela sua natureza, complementaridade, quer pela sua comum característica de armazenamento.

No capítulo seguinte iremos abordar este tipo de sistemas e as suas características, centrando-nos nos aspectos de armazenamento, processamento de informação analítica com características de georreferenciação, utilizando ferramentas e tecnologias baseadas em software de open source.

## 3. SIG e OLAP utilizando Tecnologias OpenSource

### 3.1 Introdução

Desde a década de 90 que inúmeros investigadores, empresas, organizações e comunidade em geral, têm vindo a explorar as potencialidades dos SIG e, mais recentemente, a complementaridade e exploração das potencialidades das ferramentas de processamento analítico, assim como a exploração da descoberta de conhecimento em bases de dados (geográficas). Estes temas de interligação de sistemas têm sido nos últimos anos debatidos não só a nível académico como também a nível das organizações. Na realidade um *Spatial Data Warehouse* não é uma única tecnologia mas sim a combinação de duas: uma base de dados espacial e os Sistemas de Informação Geográfica (Darmont and Boussaid, 2006, Siqueira, 2009). Ambas as tecnologias já aqui foram abordadas ao longo do documento e, conforme já referimos, com o novo tipo de dados geográfico é possível acrescentar uma nova pergunta nas análises que se podem efectuar. Até então era possível fazer análises para responder a perguntas sobre “Quando”, “Quem”, “Como”. Com o acréscimo da dimensão geográfica, é possível adicionar a resposta a pergunta do tipo “onde”. Tipicamente a resposta às questões submetidas ao sistema (através de comandos SQL) é apresentada em tabelas, sendo de fácil compreensão, mas para questões com a componente geográfica a apresentação dos dados é mais amigável e intuitiva quando

apresentada através de um mapa (Silva, 2010). Na figura 14 apresenta-se uma ilustração de uma visualização de uma análise, através de um mapa de um país categorizado por um conjunto discreto de intervalo de valores monetários (neste caso valores despendidos num determinado tema, como por exemplo, fundos gastos no tratamento de uma doença, a população de um país).

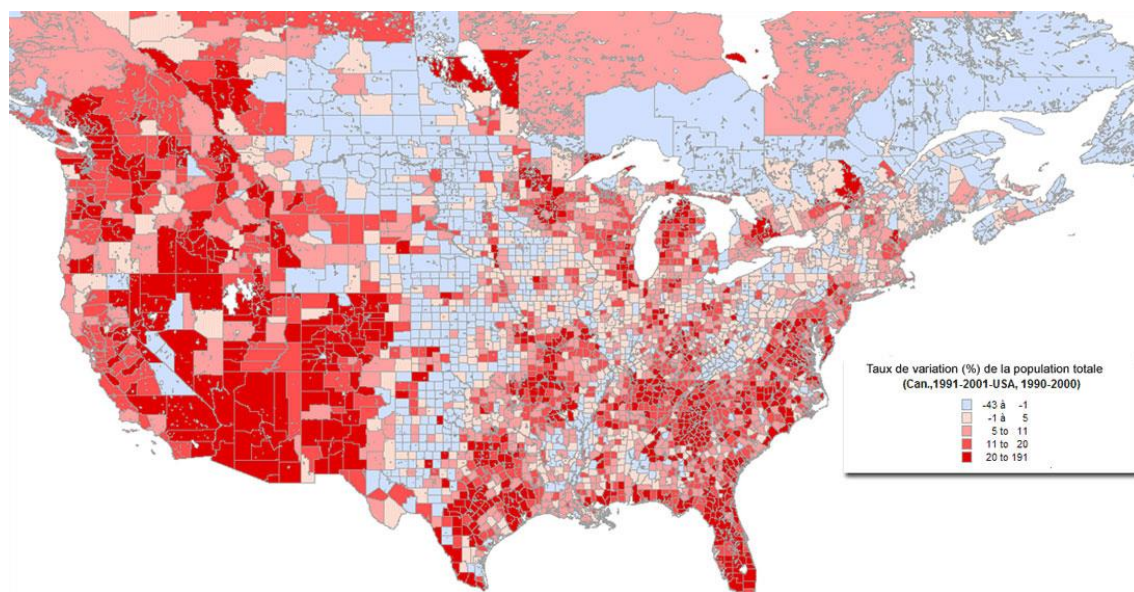


Figura 14 – Ilustração de uma visualização de Informação Geográfica com análise espacial de dados. Fonte: ([http://laser.ucs.inrs.ca/lmg/Carte\\_CanEU.jpg](http://laser.ucs.inrs.ca/lmg/Carte_CanEU.jpg))

A disponibilização destes dados georeferenciados, faz com que os sistemas sejam muito poderosos em diversas análises, para que a organização seja mais competitiva no mercado (Ferraz and Santos, 2010). Para a realização de projectos deste tipo existem diversos desafios na implementação, tais como a criação de um modelo, a integração de dados, a construção dos cubos, e por fim a geração de uma interface o mais versátil possível para os utilizadores.

Neste capítulo iremos abordar de uma forma geral, os conceitos associados à interligação entre os SIG, os SDW e OLAP, em particular a sua arquitectura de camadas. Iremos apresentar o estado da arte, quer a nível académico, quer a nível das organizações, no que se refere à utilização destes sistemas, dando especial atenção à utilização de ferramentas Open Source no processo de implementação deste tipo de sistemas.



### 3.2 Arquitectura Geral de Camadas

A construção de um Spatial Data Warehouse é, como já se referiu, a combinação de tecnologias entre as quais uma Base de Dados Espacial. Um dos principais processos na construção corresponde à modelação (modelação multidimensional). A utilização do novo tipo de dados espaciais/geográficos acrescenta vários desafios e muita discussão sobre o tema, tal como se verificou aquando da construção do modelo multidimensional de dados. Este novo tipo de dados torna o contexto mais complexo, e requer novas técnicas de representação de processamento, sendo este um dos passos mais importantes ao longo da construção e tem merecido especial atenção ao longo da última década. Será com base nesta estrutura que será assente a construção dos Cubos, das dimensões e das hierarquias das dimensões. (Tripathy et al., 2010). O estudo de Tripathy et al., 2010 apresenta uma arquitectura típica implementada em várias camadas: “Internal Layer”, “Designed Layer”, “Operational Layer”, “Display Layer”, a qual se ilustra na figura 15.

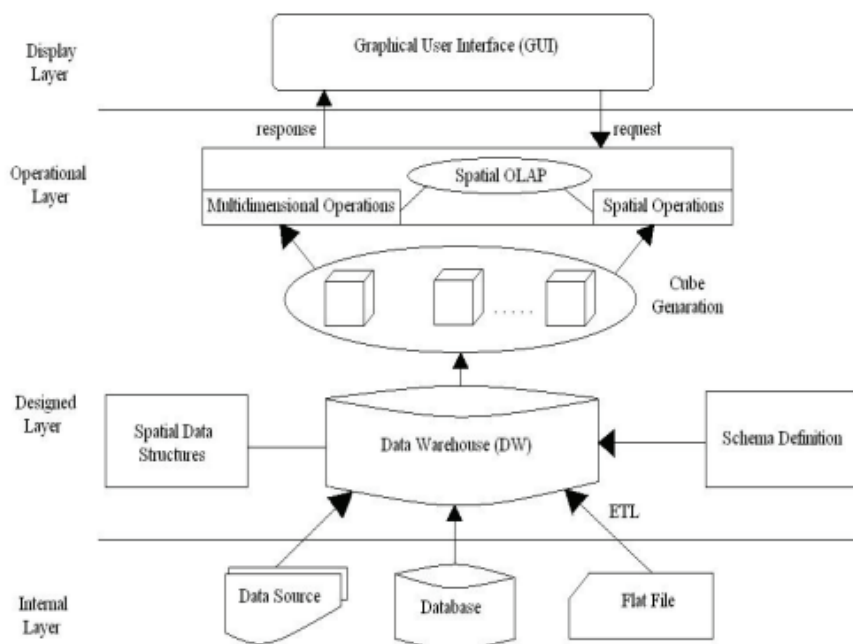


Figura 15 - Arquitectura de um Data Warehouse espacial Adaptado: (Tripathy et al., 2010)

Na primeira camada, são apresentadas todas as fontes de dados que serão utilizadas na recolha da informação. Será destas fontes de dados que o processo ETL (também designado por SETI – Seleção, Extração, Transformação e Integração de dados) carregará a informação para o Data Warehouse. A camada seguinte apresenta

o Data Warehouse sendo este modelado segundo várias perspectivas, pois como mostra a figura 15, a área responsável pelos dados espaciais encontra-se adjacente à estrutura alfanumérica e não integrada nesta.

A camada seguinte apresenta os cubos e as ferramentas que os manipulam, utilizando o Drill-Dow/Roll-Up e Slice/Dice nas Dimensões e hierarquias alfanuméricas e espaciais. Por fim a última camada corresponde a interface gráfica do utilizador, ou seja, o modo como a informação é apresentada ao utilizador final. Nesta fase a apresentação pode ser muito específica dependendo da área de negócio. Com o uso da informação geográfica é possível a utilização de mapas para a apresentação dos dados o que os torna mais apelativos, e mais fáceis de entender.

### 3.3 Estado da Arte

#### 3.3.1 A Nível Académico

Neste capítulo serão apresentadas as considerações a nível académico sobre os Sistemas de Processamento Analítico integrados com Sistemas de Informação Geográfica. A construção de um *Spatial Data Warehouse* (SDW) é, como já se referiu, a combinação de tecnologias entre as quais uma Base de Dados Espacial e os Sistemas de Informação Geográficos (SIG), ilustrando-se na figura 16 a relação entre as tecnologias referidas.

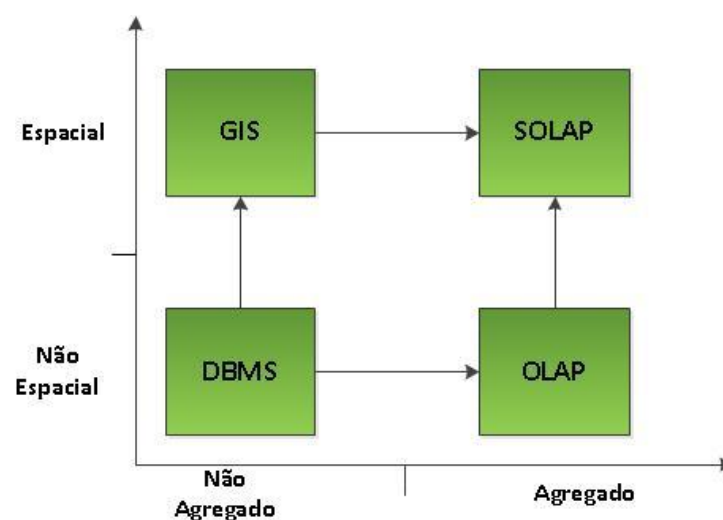


Figura 16 – Esquema geral associado aos Sistemas OLAP com e sem tratamento de informação geográfica (2011)

Em primeiro e a mais antiga, o Sistemas de Gestão de Base de Dados (SGBD), evoluíram individualmente, mas podem-se complementar à uma a outra. O SGBD estendeu-se para permitir o OLAP, e os Data Warehouse tradicionais, que conhecemos e que permite uma melhor análise dos dados. Actualmente está a surgir a extensão das funcionalidades dos SIG para OLAP, permitindo ter o que se denomina de SOLAP (Spatial Online Analytical Processing).

Para se criar um Data Warehouse com ou sem informação geográfica, são levantados vários desafios, que se podem dividir em quatro pontos: modelo de dados; o processo ETL; a tarefa de processamento e construção das estruturas multidimensionais de dados (ou cubos de dados) e, o processo de análise de dados.

A construção do modelo de dados é, naturalmente, um passo fundamental, para o desenvolvimento de um SDW. A construção de um modelo para suportar um SDW, é um processo complexo, pois tem como objectivo “juntar” dados de várias fontes. Estes modelos para os sistemas tradicionais, ou seja, somente com dados alfanuméricos, têm sido amplamente investigados. A grande maioria dos sistemas comerciais ainda não contempla a capacidade de receber estes novos dados georeferenciados. A utilização do novo tipo de dados espaciais/geográficos acrescenta vários desafios. Este novo tipo de dados torna o contexto mais complexo, e requer novas técnicas de representação de processamento. Este é um dos passos mais importantes ao longo da construção e tem merecido especial atenção ao longo da última década (Taniar, 2009).

Como as implementações deste tipo de solução ainda são reduzidas, faz com que ainda não se tenha definido standards, sendo que cada organização necessita de optar sempre pela melhor solução para o seu negocio. Por isso, vários autores apresentam vários modelos para responder as várias situações que este novo tipo de dados apresenta.

Na secção 2.3.3 apresentamos a modelação dimensional de dados presentes nos SDW de armazenamento de dados alfanuméricos. De seguida centramo-nos nas propostas de criação de modelos assentes nos Spatial Data Warehouse para armazenamento e processamento de informação geográfica. A primeira proposta para um modelo com características espaciais foi desenvolvida por Han et al. no qual foram introduzidos dois novos conceitos: uma dimensão espacial e as métricas igualmente espaciais. Segundo Darmont and Boussaid (2006), estes dois novos conceitos permitem que se consiga análises a diferentes níveis de granularidade. Nesta granularidade encontra-se então o novo tipo de dados que a diferentes níveis se pode apresentar como pontos, linhas ou polígonos.

Em 2001, Rivest et al., apresentou um modelo que estende a definição de métrica espacial, que se apresentou no modelo anterior. Esta alteração fez com o que os números de operadores topológicos aumentassem. Por esta altura, eram vários os autores que reivindicavam o aumento de soluções para apresentar ao utilizador final. Foi aqui que Marchand et al., (2003), definiu um tipo de dimensão, utilizando os diferentes operadores topológicos para os dados espaciais, consoante o nível de detalhe que se pretendia. Assim os dados não ficavam só particionados pelas dimensões, mas também pelas relações topológicas existentes. No entanto também em 2001, Shekhar et al., propôs o “map cube”, ou seja, a extensão das agregações efectuadas aos dados alfanuméricos, à informação geográfica.

Jensen et al., (2002) apresentou um requisito muito importante para as aplicações capazes de interpretar a informação geográfica. O autor propôs de forma conceptual um modelo que possibilitasse a definição de dimensões, cujos níveis estão relacionados, através de uma “contenção parcial”. Um exemplo desta contenção parcial é uma linha de caminho-de-ferro que atravessa uma determinada região. Esta contenção pode ainda conter um grau, por exemplo, a linha pode ser definida como 0,75 numa determinada região. Este foi o primeiro modelo a considerar incertezas nos dados, o que é muito significativo em aplicações reais.

Fidalgo et al., (2004), apresentou o GeoDWFrame baseado no modelo em estrela, referido anteriormente. Este modelo contudo não prevê as métricas espaciais. Aqui as dimensões são classificadas em termos de complexidade, como ilustra a figura 17.

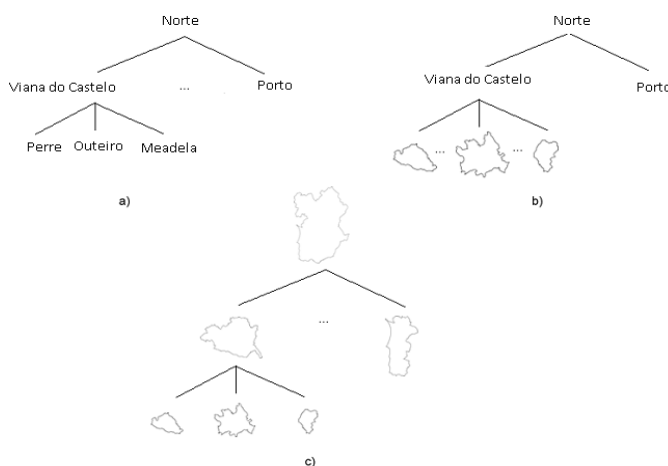


Figura 17 – Exemplificação de hierarquias de visualização de informação geográfica.

Durante o mesmo ano (Malinowski and Zimányi, 2007) apresentaram uma abordagem diferente, o SMD baseado no paradigma da Entidade Relacional. O desenho

básico desta abordagem são os factos e as dimensões. Cada entidade pode possuir um ou vários tipos de entidades, que podem ser do tipo geométrico. Os factos representam a *n-ésima* relação com os diferentes níveis das dimensões, que culminam com as métricas podendo ser ou não espaciais. Todas estas relações estão associadas a diferentes granularidades, logo temos agregações que fazem sumários de dados aos diferentes níveis. Segundo Darmont and Boussaid, 2006, foi em 2005 que se formalizou a primeira classificação de uma agregação tempo-espacial por López et al., 2005.

Contudo, qualquer que seja o modelo escolhido, deve obedecer a três princípios: **Desempenho, Flexibilidade, Extensibilidade** (Baltzer, 2011).

- **Desempenho:** uma arquitectura modular e uma implementação permite a utilização de algoritmos de alto desempenho, quer em processamento de dados quer em consulta de dados.
- **Flexibilidade:** a arquitectura deve permitir reajustes ao longo do tempo, ou seja, estar preparada para mudar de estratégia/lógica, para que o desempenho aumente.
- **Extensibilidade:** esta deve estar preparada para que novas funcionalidades sejam implementadas de uma forma fácil, e que o desempenho seja alto.

Neste contexto de interligação entre os SIG e as ferramentas OLAP é agora possível fazer análises com informação geográfica, nomeadamente operações de “Roll-up” e “Drill-Down” como se ilustra na figura 18.

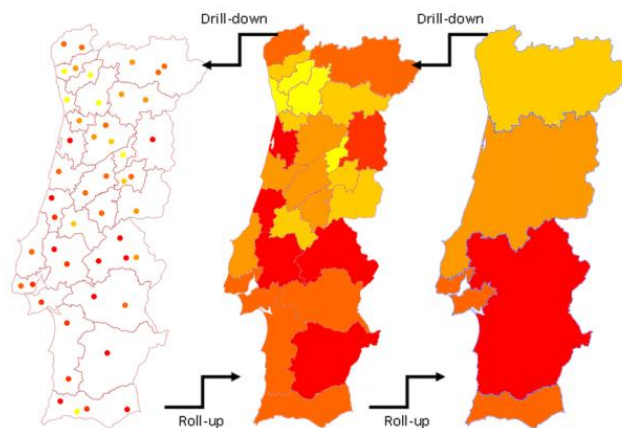


Figura 18 – Exemplificação do resultado de operações OLAP usando dados com informação geográfica (Silva, 2010).

### **3.3.2. Implementações realizadas**

Tal como já referimos, o volume de mercado deste tipo de soluções apesar de estar a crescer ainda é reduzido em comparação com os *Data Warehouses* “tradicionais”. No entanto as organizações já perceberam a potencialidade deste tipo de soluções e já estão a ser aplicados em diversas áreas de negócio. Podemos sempre encontrar um exemplo de um DW “tradicional” que contém as vendas da organização com os dados alfanuméricos “tradicionais” e que é estendido para suportar do tipo espacial para permitir novas análises.

As primeiras implementações que se realizaram tinham poucas preocupações a nível técnico em questões de desempenho e armazenamento. Estas duas componentes são muito importantes pois o objetivo da construção de um sistema deste tipo é a rapidez na análise dos dados. Quanto ao armazenamento também deve ser devidamente cuidado, pois o volume de dados pode aumentar significativamente com o decorrer do tempo e o sistema pode entrar em colapso. O objetivo é então controlar estas duas variáveis com o máximo de precisão possível para que o produto seja escalável, ou seja, a necessidade de pensar no futuro, com o aumento do volume de dados o desempenho se irá manter nos padrões aceitáveis.

Este tipo de informação georreferenciada esta actualmente a ser utilizada em vários contextos tal como se pode analisar em Wang and Zhang, 2010, Lianbo et al., 2010, Chen et al., 2011, Böhm et al., 2011, Matias and Moura-Pires, 2005, Anderson et al., 2002a, Anderson et al., 2002b, Cheng-fang et al., 2010, Ying et al., 2012, Bansal, 2010, em particular quando aplicadas às seguintes áreas: acidentes rodoviários, ocupação de solos agrícolas, distribuição energia, saúde pública, poluição, prospeção de petróleo, planeamento de construção civil, entre outras. Na figura 19 ilustra-se um exemplo de um SIG.

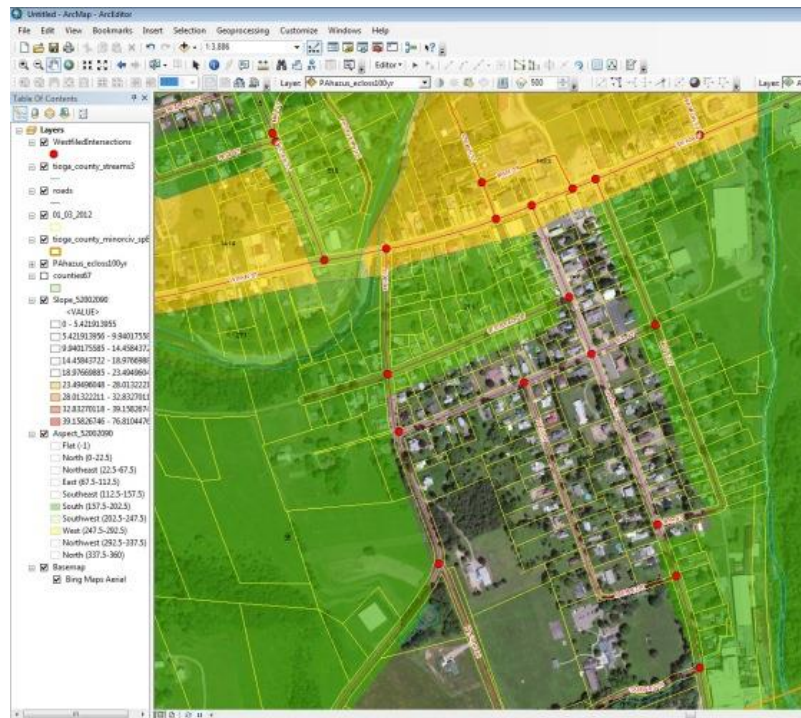


Figura 19 - Exemplo de um SIG. (Wildlife)

Mais uma vez se consegue identificar a resposta para a pergunta “Onde?” quando iniciava o capítulo. É possível determinar qual a curva onde ocorreram mais acidentes, onde se deve cultivar cereais (milho, trigo), onde seria a melhor localização para uma barragem, onde seria mais devastador o efeito de uma epidemia, onde se sentiria mais os efeitos da poluição (por exemplo os efeitos da radiação).

### 3.3.3. Ferramentas SOLAP

Actualmente existem no mercado diversas ferramentas para a construção deste tipo de soluções. Ao longo deste tópico serão abordadas somente ferramentas para todas as fases da construção excepto ao nível da base de dados que já foi apresentado anteriormente. Por sua vez, a actual situação económica/financeira das organizações é um entrave a que estas invistam em Sistemas de Informação pois os custos de soluções proprietárias por vezes são bastante elevados. Por esta razão as soluções aqui apresentadas são todas de carácter Open Source.

Apesar deste tipo de ferramentas estar cada vez mais em uso existem poucos artigos publicados a efectuar comparações entre ferramentas (Tereso and Bernardino,

2011). Nesta secção serão apresentadas as ferramentas: SpagoBI, Pentaho, JasperSoft, GeoKettle, GeoMondrian e SOLAP Layers.

A ferramenta SpagoBI<sup>12</sup> (figura 20) disponibiliza diversas funcionalidades sendo as mais robustas a criação de relatórios/gráficos, a possibilidade de construir o Processo ETL, a análise OLAP bem como Data Mining. Para o desenvolvimento dos relatórios as ferramentas utilizadas são o Eclipse BIRT, JasperReport e BusinessObjects. Para o motor OLAP estão envolvidas JPivot/Mondrian, JPalo/Mondrian e JPivot/XMLA (Tereso and Bernardino, 2011).

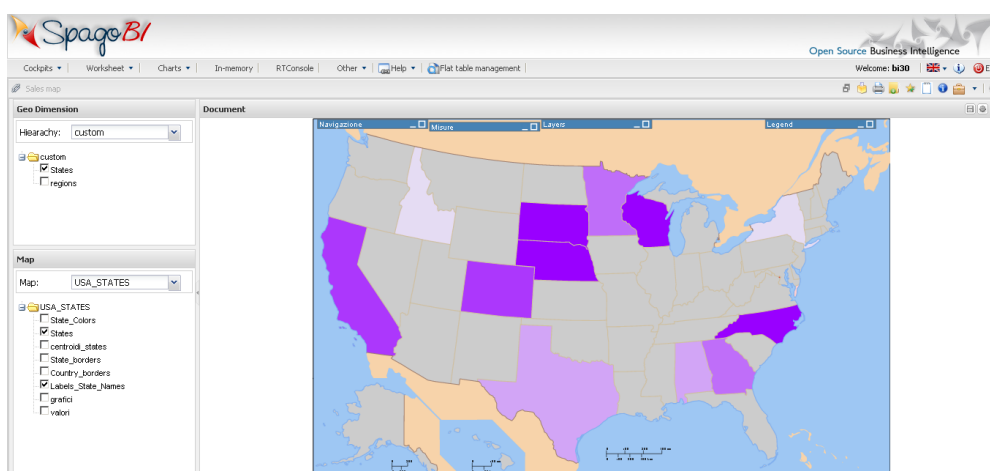


Figura 20 – Ilustração de uma ferramenta OLAP (SpagoBI ) a operar sobre Informação Geográfica.

O Pentaho<sup>13</sup> (figura 21) está disponível em duas versões a *Community* e a *Enterprise*. A primeira versão tem carácter Open Source e é suficiente na maioria dos casos para as pequenas e médias empresas. Esta permite a construção de Dashboards, de análises OLAP. O motor OLAP envolvido é o Mondrian e Ramsetcube, sendo o módulo de Data Mining da responsabilidade da ferramenta Weka.

<sup>12</sup> <http://www.spagoworld.org/xwiki/bin/view/SpagoBI/>

<sup>13</sup> <http://www.pentaho.com/>



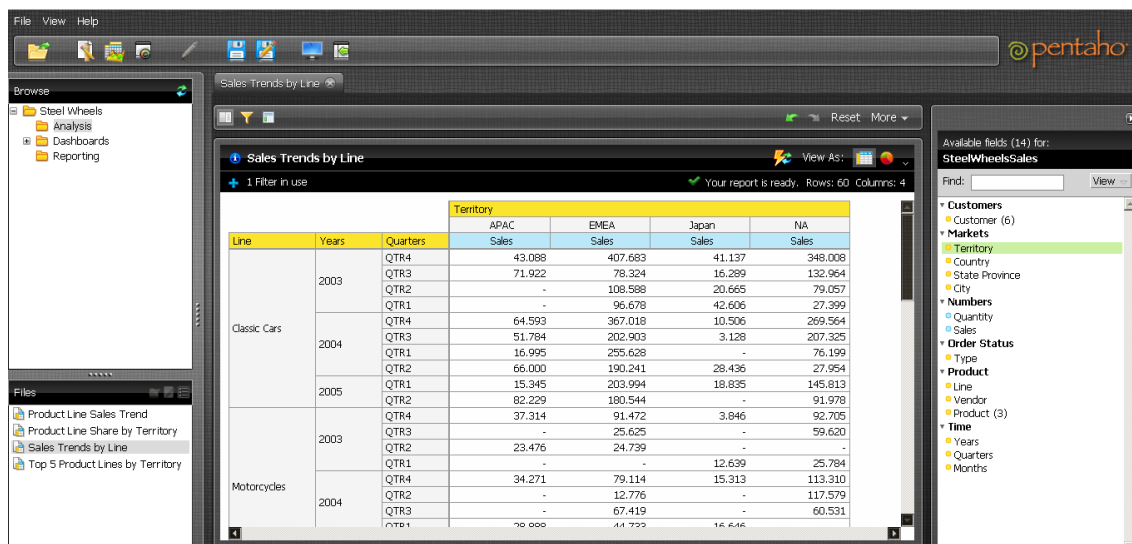


Figura 21 - Exemplo de análise alfanumérica no Pentaho.

No âmbito deste projecto e depois de analisar várias ferramentas optou-se pelas seguintes, para o desenvolvimento do protótipo.

- **GeoKettle**<sup>14</sup>: É uma aplicação capaz de integrar informação de vários sistemas, num único sistema. Com esta ferramenta é possível realizar operações com os dados para a sua perfeita integração. Tem a capacidade de usar informação geográfica, usando para tal o formato de Shapefile ou através das pesquisas em SQL.
- **GeoMondrian**<sup>15</sup>: É um motor de OLAP espacial. É na realidade o primeiro motor SOLAP de verdade (15). Este permite a execução de pesquisas MDX (linguagem utilizada para efectuar pesquisas sobre os Cubos), também estas com capacidade de devolução de dados geográficos. Actualmente só suporta base a de dados PostgreSQL com a extensão PostGIS.
- **GeoWorkbench**<sup>16</sup>: Esta ferramenta consiste em uma ferramenta capaz de construir os cubos. Apresenta uma interface gráfica para criar e testar schemas de cubos. Tem a capacidade de processar pesquisas MDX com o ROLAP (Relational OLAP). Os ficheiros criados nesta são do tipo XML com um formato específico de modo a que o GeoMondrian os consiga interpretar.
- **JPivot**<sup>17</sup>: É uma biblioteca que permite a representação gráfica de informação obtida num OLAP, permitindo efectuar as ações de slice-dice, drill-down e roll-up.

<sup>14</sup> <http://www.spatialytics.org/projects/geokettle/>

<sup>15</sup> <http://www.spatialytics.org/projects/geomondrian>

<sup>16</sup> <http://mondrian.pentaho.com/documentation/workbench.php>

<sup>17</sup> <http://jpivot.sourceforge.net/>

- **OpenLayers:** É uma biblioteca inteiramente desenvolvida em JavaScript , que permite a visualização de mapas nos browsers, sem qualquer tipo de dependências no servidor. Tem uma API (*Application Programming Interface*) para a construção de carácter geográfico, semelhante ao GoogleMaps ou MSN Virtual Earth.
- **Ext JS**<sup>18</sup>:É também uma biblioteca totalmente desenvolvida em Javascript, disponibilizando uma framework para aumentar a interatividade das aplicações WEB, usando programação em AJAX, DHTML e DOM.
- **Apache/Tomcat**<sup>19</sup>: Apache é um servidor web de carácter Open Source com suporte a sistemas com ou sem informação georreferenciada.

### 3.4 Reflexão Crítica

Nesta seção apresentamos os conceitos base associados aos Sistemas de Data Warehousing, como a modelação dimensional, as suas características e terminologias (ex.: dimensões, atributos, medidas, tabela de fatos, entre outras). Abordamos a complementaridade dos tradicionais sistemas de processamento analítico com os Sistemas de Informação Geográfico.

Neste contexto, foram identificadas várias ferramentas baseadas em tecnologia Open Source orientadas a realizar todas as etapas necessárias à construção de um spatial data warehouse. Apresentaram-se as ferramentas SpagoBI, Pentaho, GeoKettle, GeoMondrian, GeoWorkbench, JPivot e OpenLayers.

Com base no estudo descrito neste capítulo, concluímos em primeiro lugar que, o processo de disponibilização de informação analítica com informação geográfica requer um forte conhecimento dos conceitos base do desenho um Sistema de Data Warehousing e conseqüentemente da semântica associada aos sistemas de processamento analítico. Em segundo lugar, constata-se que, nos últimos anos, há uma forte aposta por parte da comunidade que defende a utilização do software open source em aliar as vantagens associadas a estes tipos de sistemas, tipicamente aplicados a informação alfanumérica, na sua integração com os Sistemas de informação geográfica. Isto reflete-se no aparecimento de um alargado conjunto de ferramentas. Conclui-se que as ferramentas SpagoBI e o Pentaho, não reuniam as funcionalidades necessárias para

---

<sup>18</sup> [http://en.wikipedia.org/wiki/Ext\\_JS](http://en.wikipedia.org/wiki/Ext_JS)

<sup>19</sup> <http://tomcat.apache.org/>

a realização deste projecto. Com isto foi necessário uma nova abordagem para a apresentação dos dados aos utilizadores, foi pensada/decidida a elaboração de uma interface Web independente destas ferramentas.

No capítulo seguinte será apresentado um caso de estudo, apresentando um protótipo de modo a expor todas as vantagens e problemas na implementação de um spatial data warehouse.



## 4. Caso de Estudo

### 4.1 Introdução

Ao longo deste capítulo serão abordadas todas as acções necessárias para a realização de um Sistema de Data Warehousing Espacial. O objectivo do projecto tal como já se referiu, é estudar, experimentar e demonstrar a possibilidade da realização de um SDW com informação geográfica utilizando ferramentas baseadas na filosofia open source assim como analisar o seu desempenho em termos de processamento. Tal como se apresentou no capítulo inicial pretende-se que este estudo seja exploratório, não procurando validar relações de causalidade nem verificar uma teoria. Na realidade, pretende-se identificar preocupações, dificuldades e hesitações na criação de sistemas que tirem partido dos tradicionais Sistemas de Processamento Analítico e dos Sistemas de Informação Geográfica.

### 4.2 Características da Informação

Os dados utilizados para a realização do presente caso de teste, são relativos aos censos 2011 realizados em Portugal pelo Instituto Nacional de Estatística<sup>20</sup>. Todos estes dados estão disponíveis a qualquer cidadão a partir do *site* do INE, na sua secção

---

<sup>20</sup> <http://www.ine.pt>

associada aos censos<sup>21</sup>. Os dados estão divididos em três áreas: Portugal Continental, a Região Autónoma da Madeira e a Região Autónoma dos Açores, sendo utilizados no estudo apenas os dados correspondentes a Portugal Continental. Na figura 22 ilustra-se um esquemático com o tipo de dados disponibilizados pelo INE, em que, além do formato standard de texto separado por ponto e vírgula, os ficheiros CSV, é também disponibilizada a informação nos formatos descritos na secção 2.2.3, nomeadamente no formato shapefile.

A escolha deste tipo de conjunto de dados para este caso de estudo deveu-se ao facto de serem dados reais de carácter público para qualquer cidadão. Outro ponto importante foi o grande volume de dados devido ao âmbito do projecto incidir sobre o desempenho do software usado. Na figura 22 ilustra-se um exemplo de categorização de informação disponibilizada pelo INE, nomeadamente por concelhos.

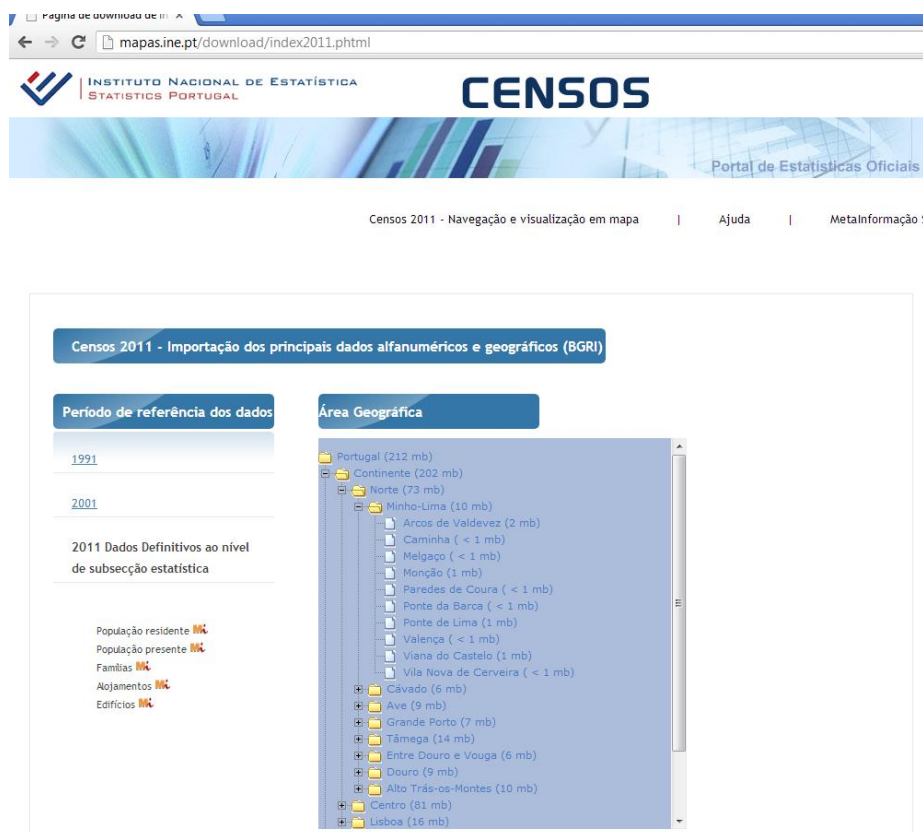


Figura 22 – Tipo de informação disponibilizada pelo INE.

<sup>21</sup> <http://mapas.ine.pt/download/index2011.phtml>.

### 4.3 Arquitectura Geral e Ferramentas utilizadas

As ferramentas utilizadas no estudo foram as descritas no capítulo anterior, nomeadamente, Apache/Tomcat, GeoKettle, GeoMondrian, GeoWorkbench, JPivot e OpenLayers. Em termos de implementação física, o caso de estudo foi totalmente construído através da virtualização de uma máquina com as características descritas na figura 23. Este ponto é importante, pois tal como foi referido presente trabalho abrange o desempenho das ferramentas utilizadas. A máquina virtual serviu de servidor e de cliente, no entanto para uma implementação real, recomenda-se a separação em servidor e cliente, para um melhor desempenho.

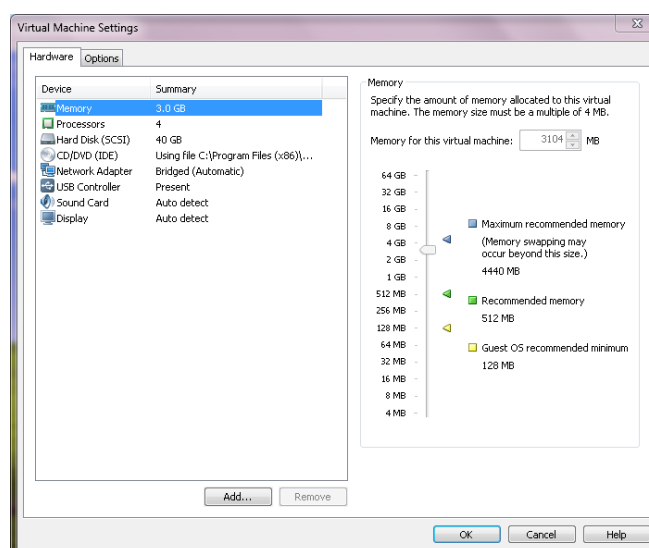


Figura 23 - Características da máquina virtual.

A arquitectura do sistema de suporte web à visualização dos dados analíticos assenta numa estrutura de três camadas com base num servidor Web, como por exemplo o Apache/Tomcat. Em termos de camada de dados, a informação está armazenada numa base de dados PostgreSQL com a componente geográfica PostGIS para armazenamento dos dados no formato vectorial. Em termos de camada lógica de negócio e, como implementamos um sistema protótipo de visualização de informação analítica com características geográficas as ferramentas utilizadas foram: o GeoKettle, GeoMondrian, GeoWorkbench e JPivot. Com estas ferramentas foi construído o processo ETL para carregar o DW, e a construção do modelo multidimensional. As quatro ferramentas referidas são suportadas pela tecnologia Java. Na camada de apresentação dos dados utilizamos o ExtJS, GeoExt e o OpenLayers. A figura 24 seguinte ilustra a arquitectura que foi seguida ao longo de todo o protótipo.

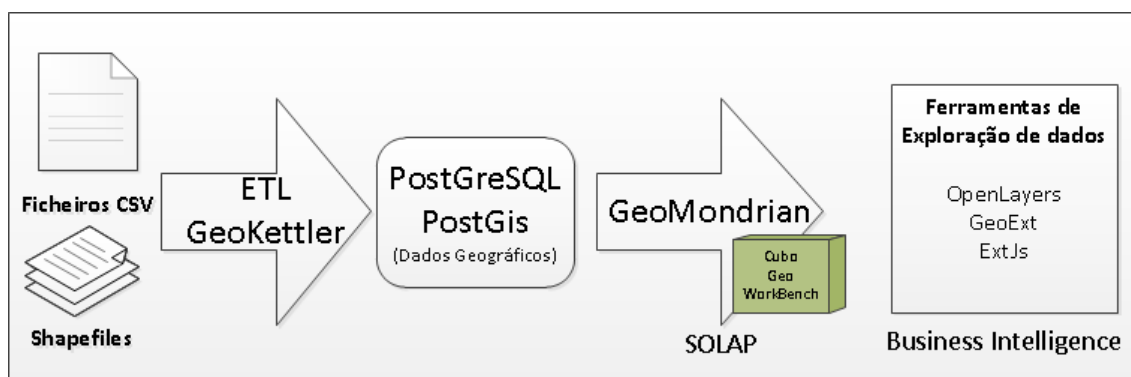


Figura 24 – Esquema da arquitectura geral do sistema.

## 4.4. Tarefas realizadas

As tarefas realizadas neste projecto seguiram uma metodologia tradicional. Em primeiro lugar procedeu-se ao estudo dos conceitos e terminologias associadas a este tipo de sistemas, assim como à instalação das ferramentas. Depois, passamos para a construção do modelo multidimensional de suporte aos dados e conseqüentemente ao desenvolvimento do processo ETL para carregamento do Data Warehouse. De seguida procedemos à construção do modelo OLAP e ao desenvolvimento de uma interface gráfica para a visualização dos dados.

### 4.4.1 Instalação das ferramentas

O processo de instalação iniciou-se pelas ferramentas GeoKettle (para integrar informação de vários sistemas, num único sistema) e o GeoMondrian (como motor do SOLAP), as quais, devido a terem sido desenvolvidas muito recentemente, deparamo-nos com algumas dificuldades. A principal deveu-se, efectivamente, à pouca informação oficial disponível de como efectuar este procedimento de utilização das duas ferramentas, o que levou a uma grande investigação para a resolução dos problemas.

Relativamente à instalação do GeoKettle decorreu sem problemas, pois possui um wizard que realiza todas as configurações necessárias. No entanto, aquando da primeira utilização do software verificou-se a necessidade de uma *password*, se pretendêssemos utilizar todas as funcionalidades da mesma. A utilização desta *password* disponibiliza a criação de um repositório. Este repositório é nada mais que



uma base de dados que armazena todas os cenários criados. A pesquisa efectuada sobretudo em fóruns, mostrou-se positiva e descobriu-se que a mesma era “*admin*”.

A ferramenta GeoMondrian, além da dificuldade em obter documentação, verificou-se que muita desta, a grande maioria não oficial, era “contra informação”. O primeiro problema detectado verificou-se aquando da instalação da versão disponibilizada na página oficial do produto. A instalação deste mostrou que o produto não funcionava, o que originou uma nova pesquisa. Esta pesquisa consumiu bastante tempo, o qual se fez sentir em fases mais adiantadas do projecto. Uma das abordagens na pesquisa, passou por procurar informação acerca da aplicação Mondrian, que tem o mesmo propósito mas sem a capacidade de processar informação geográfica. A instalação desta consiste em publicar um ficheiro WAR num servidor Apache/Tomcat e aceder ao *path* a partir do browser.

#### **4.4.2 Construção do Modelo Multidimensional**

Tal como se referiu no estado da arte, a construção deste tipo de modelos ainda se encontra em debate, o que origina várias abordagens sobre o tema. O modelo usado no projecto, foi elaborado como um modelo em estrela, em que todas as dimensões são de carácter híbrido, ou seja, possuem dados alfanuméricos e geográficos.

Um das melhores práticas na construção de um modelo multidimensional é a construção de uma dimensão tempo. O presente projecto não se colocou essa dimensão no modelo pois os dados referiam-se a um mesmo período de tempo. No caso de este modelo ser escalado para a integração dos vários censos realizados em Portugal, era necessário a sua implementação. O modelo multidimensional ilustrado na figura 25 é composto por quatro dimensões, que passo a citar: dimensão Município; dimensão Freguesia; dimensão Secção e dimensão Subsecção. Estas dimensões encontram-se distribuídas por ordem de grandeza da informação, sendo o modelo também constituído por uma tabela de factos denominada por “*f\_census*”. Neste sentido, a informação ao nível mais baixo disponibilizado pelo INE, é a Subsecção. Adicionalmente, em todas as tabelas de dimensão é possível identificar o atributo geográfico, pois todos têm o nome de “*the\_geom*”.

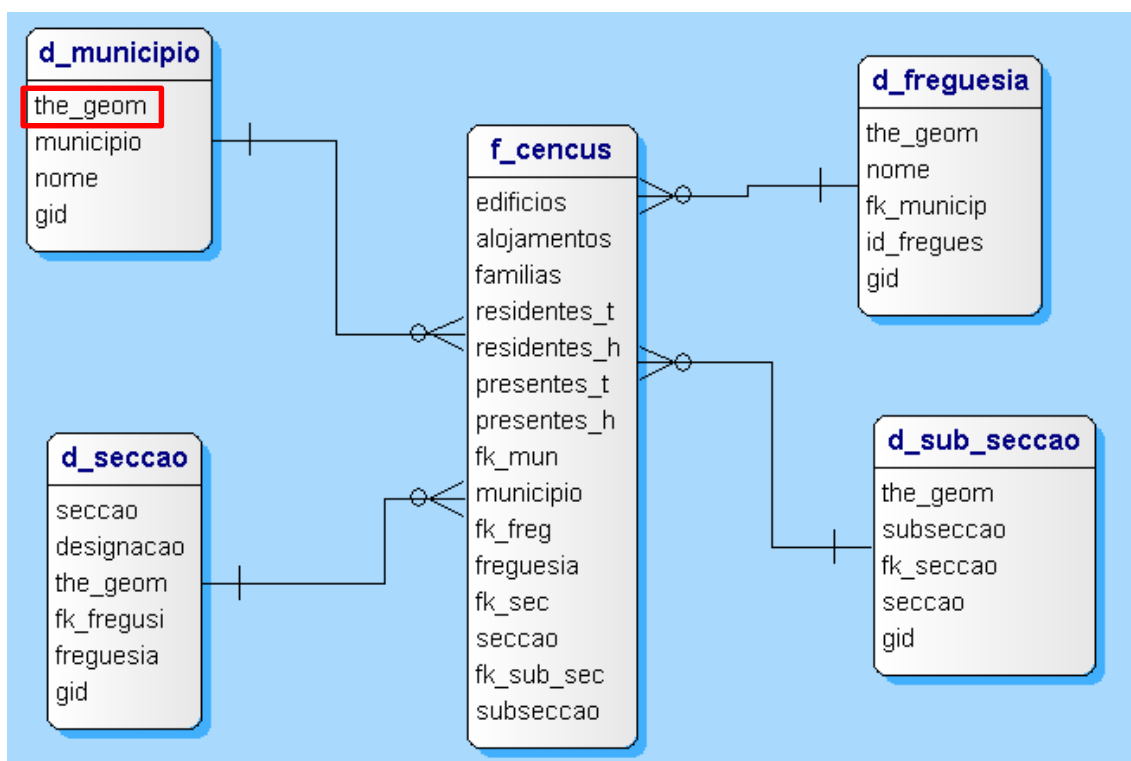


Figura 25 - Modelo Multidimensional do caso de estudo

#### 4.4.3 Construção do Processo de ETL

Tal como se referiu, os dados utilizados provém do INE, e estão disponíveis para qualquer cidadão. Os dados são disponibilizados quer em formato ficheiro CSV (o qual contém todos os dados alfanuméricos), quer através de ficheiro em formato shapefile (o qual contém todos os dados geográficos). Antes de iniciar a construção do ETL, verificou-se a granularidade dos dados, a que estes estavam disponibilizados, identificando o nível de Subsecção. Com esta informação e para facilitar o processo de ETL procedeu-se a geração de uma shapefile para cada uma das dimensões tendo em consideração o nível de agregação. Esta acção foi realizada com a ferramenta Quantum GIS, com as seguintes etapas:

1. Carregar a shapefile no Quantum GIS:

Esta acção é realizada clicando no botão que se encontra identificado pelo rectângulo vermelho na figura 26.

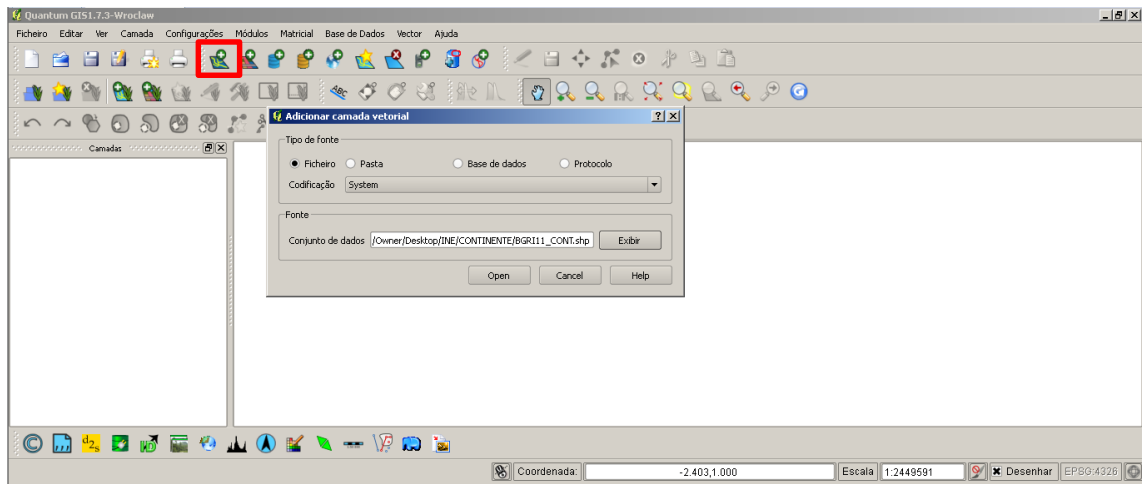


Figura 26 - Carregar ShapeFile no Quantum GIS.

2. Efectuar a acção de “Dissolver” pelo nível de agregação pretendido.

A acção de “Dissolver” encontra-se no Menu: Vector → Ferramentas de geoprocessamento → Dissolver. A opção encontra-se identificada com um rectângulo vermelho na figura 27.

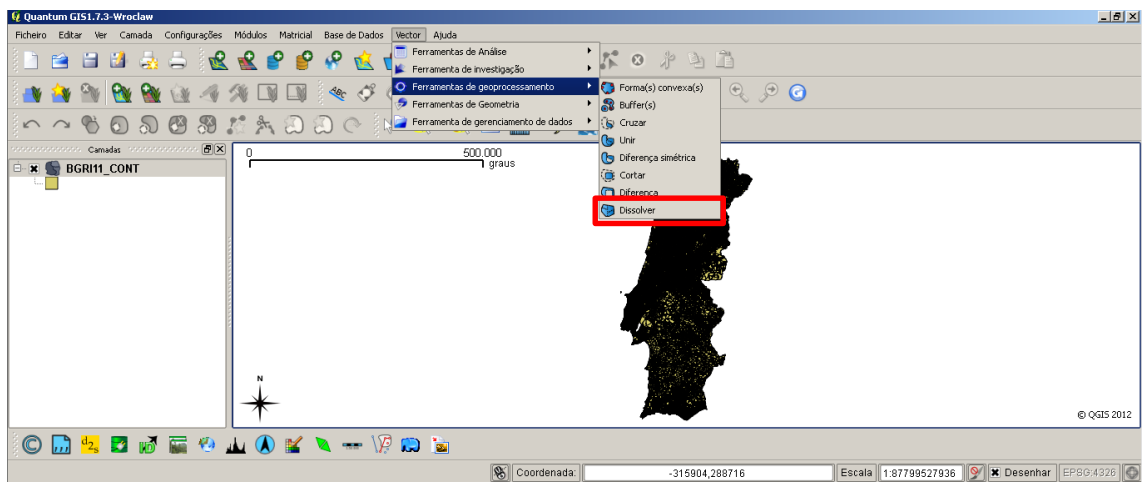


Figura 27 - Caminho da opção “Dissolver”

A etapa seguinte consiste em escolher, a shapefile que contém os dados e seleccionar a coluna pela qual pretendemos efectuar a agregação de dados (equivalente à acção de group by), tal como indica o rectângulo vermelho na figura 28.

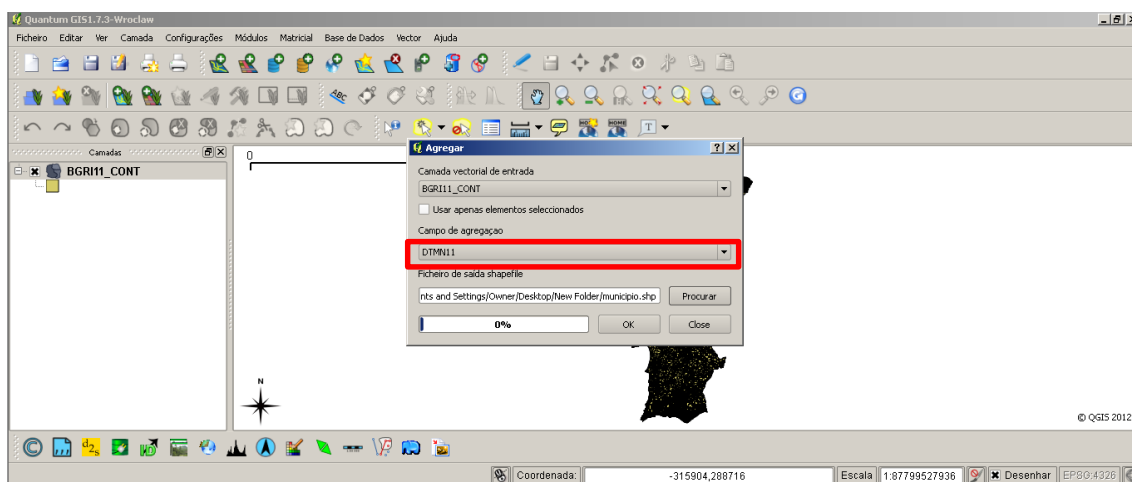


Figura 28 - Escolha da coluna para efectuar agregação de dados.

A realização desta acção é relativamente demorada, dependendo do poder de processamento da máquina em que é realizada. Este tipo de agregação de dados podia ter sido efectuado no processo de ETL, no entanto, optou-se pela sua realização de uma forma independente. Antes de apresentar o processo de ETL, apresenta-se o modo de como os dados alfanuméricos estão disponibilizados no ficheiro CSV, tal como ilustra a figura 29. O ficheiro contém a informação sobre: Edifícios; Alojamentos; Famílias; Residentes\_T (Total de Residentes); Residentes\_H (Total de Residentes do sexo masculino); Presentes\_T (Total de Presentes no local) e Presentes\_H (Total de presentes no local do sexo masculino).

```

ano;geog_2002;designacao;nivel;ordem;EDIFICIOS;ALOJAMENTOS;FAMILIAS;RESIDENTES_T;RESIDENTES_H;PRESENTES_T;PRESENTES_H
2011;PT;Portugal;Pais;1000;3543595;5877991;4048932;10561614;5047387;10281794;4869537
2011;11;Minho-Lima;Nuts III;4000;120882;150894;89681;244836;114352;236473;109173
2011;164;Pinhal Interior Norte;Nuts III;127170000;85640;96871;52026;131371;62592;126205;59391
2011;113;Ave;Nuts III;23021000;157557;221632;174632;511737;247062;497824;238361
2011;165;Dão-Lafões;Nuts III;136132000;145939;174167;104844;277216;131585;268105;125549
2011;161;Baixo Vouga;Nuts III;96347000;149860;202430;145424;390840;187085;380056;180192
2011;185;Lezíria do Tejo;Nuts III;255711000;107083;137706;97564;247449;119178;238474;113493
2011;169;Beira Interior Sul;Nuts III;162768000;46038;60204;31891;75026;35705;73077;34429
2011;162;Baixo Mondego;Nuts III;107096000;128094;193744;130942;332306;156650;329372;152603
2011;168;Beira Interior Norte;Nuts III;154815000;74400;85772;41965;104403;49550;100768;47532
2011;171;Grande Lisboa;Nuts III;192194000;277201;1066754;836648;2042326;961388;1982601;927185
2011;166;Pinhal Interior Sul;Nuts III;148694000;30617;33225;16841;40705;19328;38497;18093
2011;115;Tâmega;Nuts III;52012000;197885;248475;182824;550469;268046;527736;251736
2011;181;Alentejo Litoral;Nuts III;231884000;53448;68767;40572;97895;48610;95346;47453
2011;117;Douro;Nuts III;72951000;119390;140276;78714;205902;98450;200334;94945
2011;150;Algarve;Nuts III;262717000;198860;379937;183022;451005;220000;462825;226102
2011;16C;Médio Tejo;Nuts III;182653000;107287;137708;87284;220660;105227;212112;100140
2011;172;Península de Setúbal;Nuts III;218239000;171519;420963;312434;779373;373595;748677;354578
2011;112;Cávado;Nuts III;12002000;124403;190602;137523;410149;196820;397361;187936
2011;163;Pinhal Litoral;Nuts III;117204000;109605;144908;100702;260924;125901;250783;119845
    
```

Figura 29 - Ficheiro CSV com dados alfanuméricos.

Após reunir toda a informação pretendida efectuou-se o processo de ETL. Para a construção destes processos, tal como já se referiu no documento, utilizou-se a

ferramenta GeoKettle. A primeira etapa foi a criação de um repositório, onde serão guardados os processos realizados. Este repositório não é mais que um conjunto de tabelas em que a ferramenta armazena informação para o seu ideal funcionamento. A figura 30 apresenta a interface de Login da aplicação.

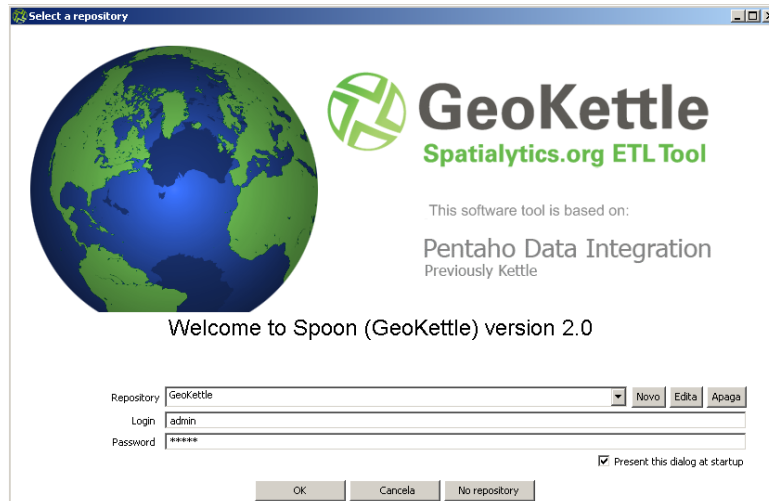


Figura 30 – Ilustração da Interface da ferramenta GeoKettle.

Tal como se indicou, o repositório é uma base de dados neste caso na mesma máquina, sendo visível no gestor de base de dados, tal como a figura 31 indica.

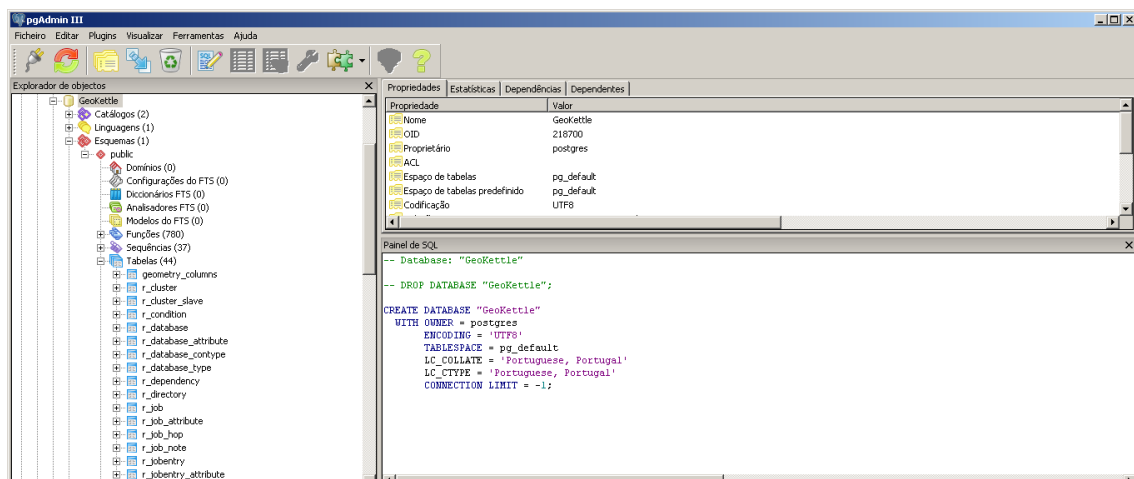


Figura 31 – Visualização do repositório (Base de Dados).

Passando para as funcionalidades, a aplicação permite então a geração de “Transformation” e “Jobs” usando as nomenclaturas da própria aplicação. Em relação à “Transformation” é possível a geração de processos de transformação de dados. Neste sentido apresenta-se uma “Transformation” utilizada no processo para carregar a

dimensão de Município, exemplificando também como proceder para carregar a mesma a partir do repositório, como ilustrado na figura 32.

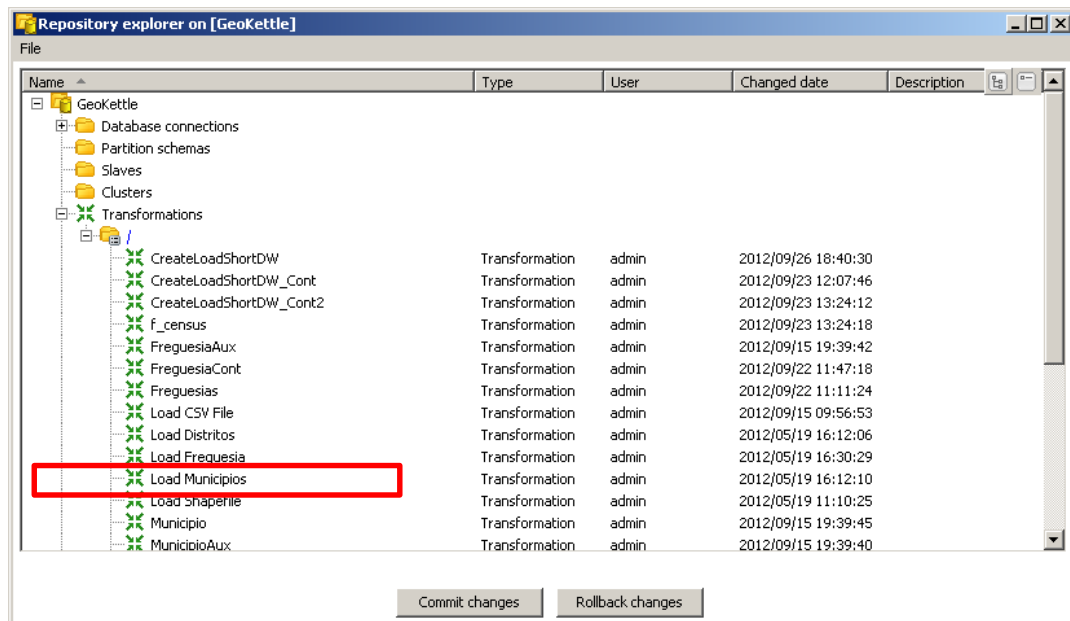


Figura 32 – Ilustração do carregamento “Tranformations” do Repositório.

O processo de carregar a dimensão é composto por duas “transformations”. A primeira serve para fazer cálculos auxiliares usando uma BD auxiliar também, e a segunda lê os dados dessa BD auxiliar para o Data Warehouse.

- *Transformation* Auxiliar

A figura 33 apresenta as etapas efectuadas na “transformations” auxiliar.

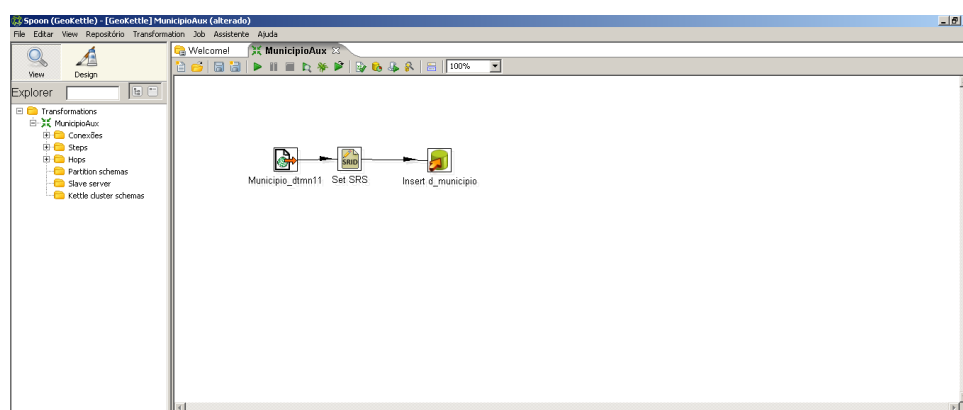


Figura 33 - Exemplificação de uma “Transformation”.

Esta é composta por três elementos, o primeiro elemento é responsável por ler a Shapefile obtida no processo de “dissolver”, referido anteriormente. O segundo é um

elemento muito importante pois garante o sistema de coordenadas pretendido, o EPSG:3763, tal como a figura 34 demonstra. Esta configuração tem de ser mantida ao longo de todo o processo de modo a garantir a consistência/coerência dos dados.

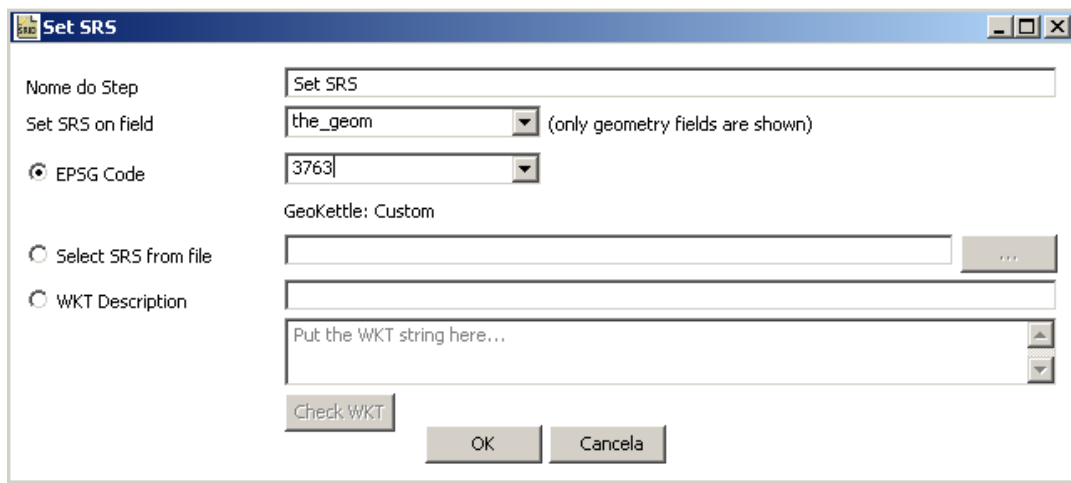


Figura 34 – Definição do sistema de coordenadas.

O último elemento garante a inserção dos dados da shapefile, com a configuração acima escolhida na referida BD auxiliar. A figura 35 identifica as colunas que serão inseridas na tabela “municipio” na Base de dados auxiliar.

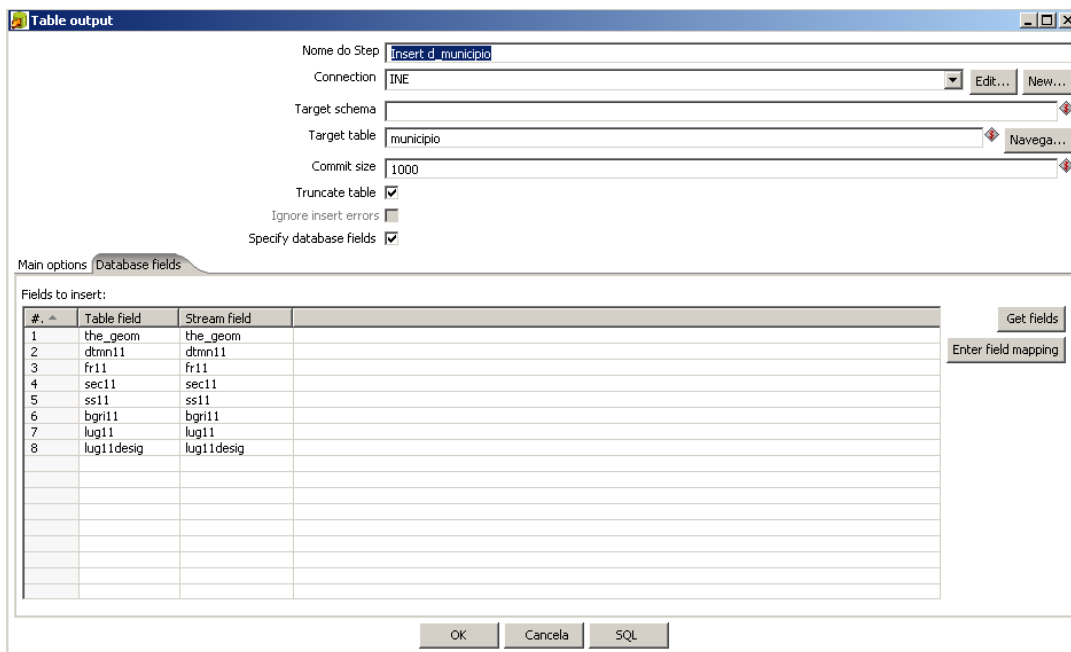


Figura 35 - Exemplificação da operação de inserir dados numa tabela, com a especificação das colunas.

- “Transformation” Final

Após a etapa anterior estar finalizada, procedeu-se a construção do processo final de carregamento da dimensão, tal como ilustrado na figura 36.

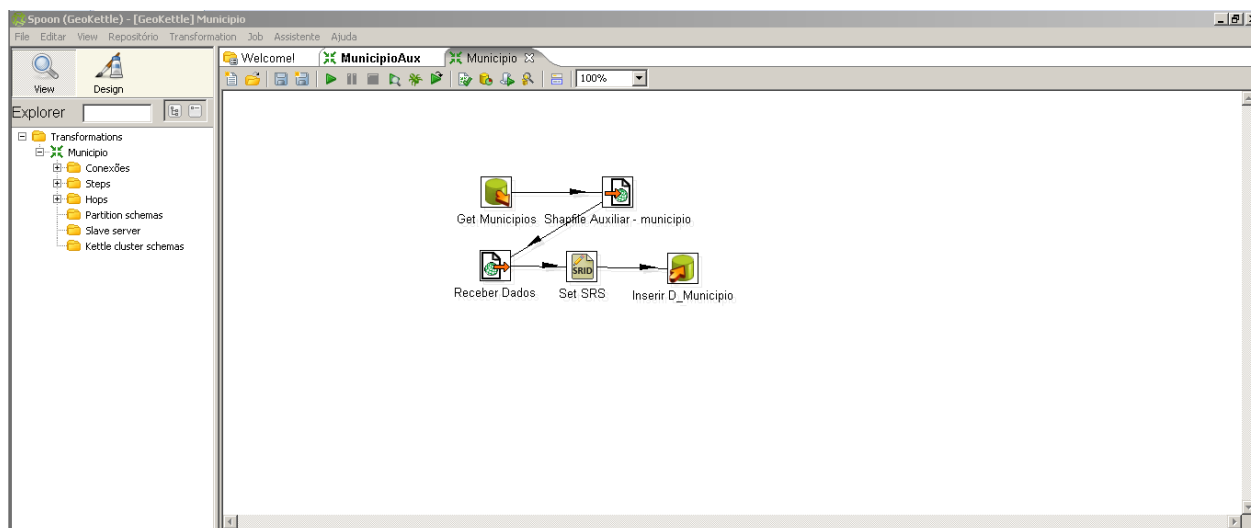


Figura 36 - Processo ETL final da dimensão Município.

Este processo tem como fases a leitura dos dados da BD auxiliar e a inserção dos mesmos no DW. Para proceder a este carregamento e por razões técnicas não conseguimos realizar directamente pois, ocorria um erro quando se tentava inserir a coluna geográfica. Para resolver esta situação, procedeu-se a geração de uma shapefile auxiliar, e consequentemente o carregamento desta, garantindo o sistema de coordenadas dos dados. Este processo de geração de shapefiles auxiliares teve um desempenho aceitável, pelo que foi implementada a mesma metodologia para as restantes dimensões.

Todos estes processos podem ser executados inúmeras vezes pois o processo garante o “truncate” da tabela destino para evitar a duplicação de dados. Se o projecto evoluísse e surgissem carregamentos com muita frequência, recomendava-se que esta abordagem fosse devidamente analisada pois o “truncate” (limpeza de todos os dados existentes na tabela), implica o total carregamento da tabela o que com grandes volumes de dados, degrada o desempenho, uma possível solução seria a implementação de um “Merge”. Numa situação destas recomenda-se a utilização de um Merge para inserir os novos registos, e actualizar os já existentes, caso se detectassem falhas nos carregamentos anteriores.



O processo seguinte, ilustrado na figura 37 refere-se ao carregamento da tabela de factos, sendo este o mais demorado devido ao volume de dados.

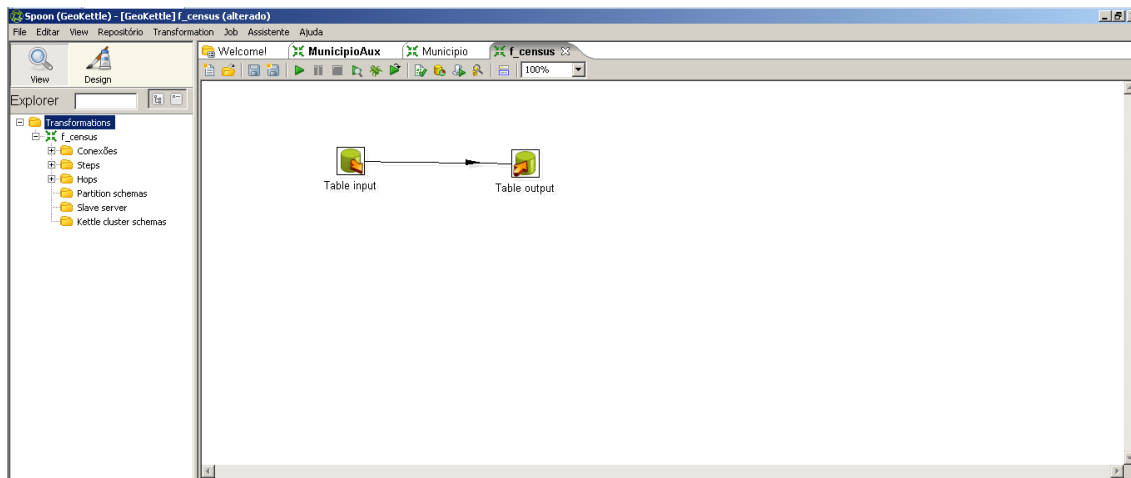


Figura 37 - Exemplificação da “Tranformations” para carregar a tabela de factos.

Para se efectuar este carregamento na tabela de factos previamente efectuou-se o carregamento do ficheiro CSV para uma tabela na base de dados auxiliar. A partir desta tabela juntamente com as tabelas auxiliares correspondentes as dimensões desenvolveu-se uma pesquisa SQL, como a figura 38 ilustra.

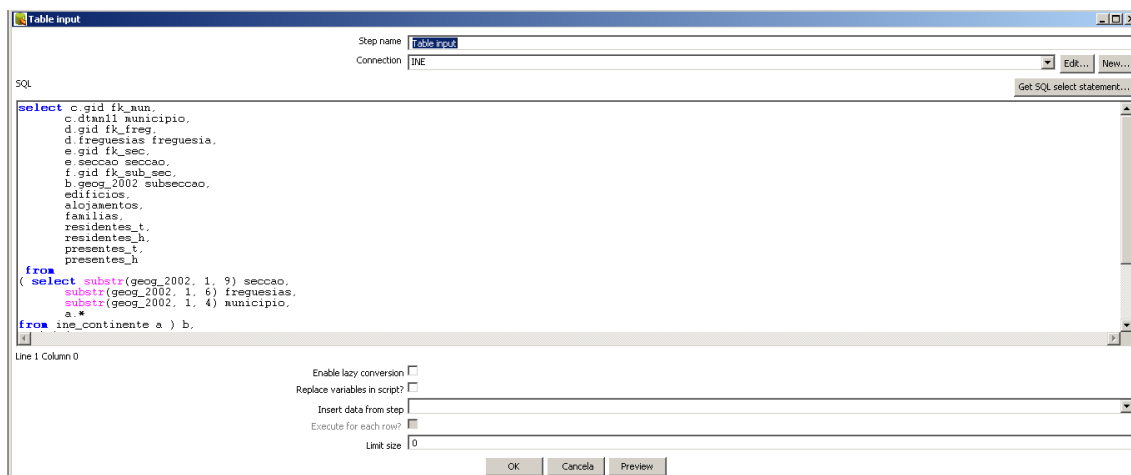


Figura 38 - Pesquisa para efectuar a selecção de dados para tabela de factos.

A aplicação disponibiliza também uma área de Debug, para visualizar como está a decorrer o processo. Mais uma vez, e como “boas práticas”, deveriam ser implementadas tabelas de registo de forma a registar todos os passos que o processo esta a executar ou já executou. No caso de estudo não foram implementadas por opção.

O processo total de ETL ronda em média uns 13 minutos, no entanto é necessário ter em conta que este tempo é obtido no ambiente de desenvolvimento, ou seja, um ambiente virtualizado com recursos mais limitados.

#### 4.4.4 Construção do modelo SOLAP

A realização deste processo foi efectuada através da aplicação GeoWorkbench. Nesta ferramenta criou-se então a estrutura do Cubo para a análise de dados com a informação geográfica. Em primeiro lugar apresenta-se as funcionalidades da aplicação que permite realizar três acções: criar um Schema (Construção dos Cubos); executar pesquisas MDX; exploração de uma Base de Dados, tal como se pode verificar na figura 39.

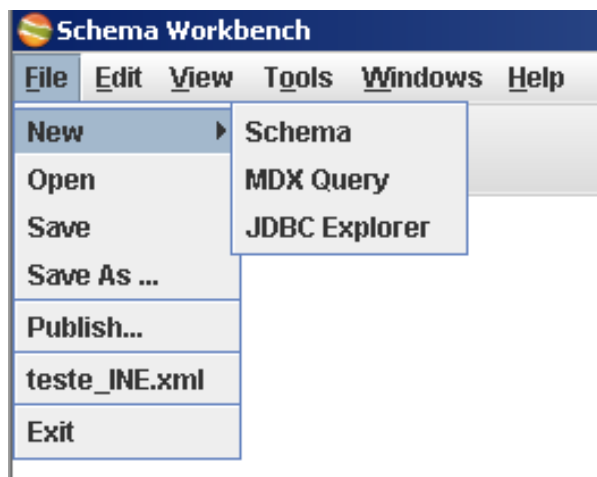


Figura 39 - Ilustração das funcionalidades disponíveis no Workbench.

A aplicação apresentou alguns problemas em termos de desempenho aquando da criação de estruturas mais complexas (Schema). Apresentava grande lentidão (alguns minutos até voltar a responder correctamente) sempre que pretendia adicionar algum campo geográfico, ou após a adição deste, quando se efectuava qualquer acção. A ferramenta começava por consumir grande parte da memória da máquina, sendo que por vezes bloqueava completamente e era necessário terminar o processo da mesma.

Um outro problema encontrado foi que algumas alterações efectuadas e posteriormente gravadas não eram reconhecidas, sendo necessário reiniciar a ferramenta. Outro problema detectado ocorreu aquando da escolha da funcionalidade

de realizar pesquisas MDX. Foi realizada uma investigação para a resolução do problema mas mostrou-se infrutífera, não se conseguindo resolver a situação identificada pela figura 40.

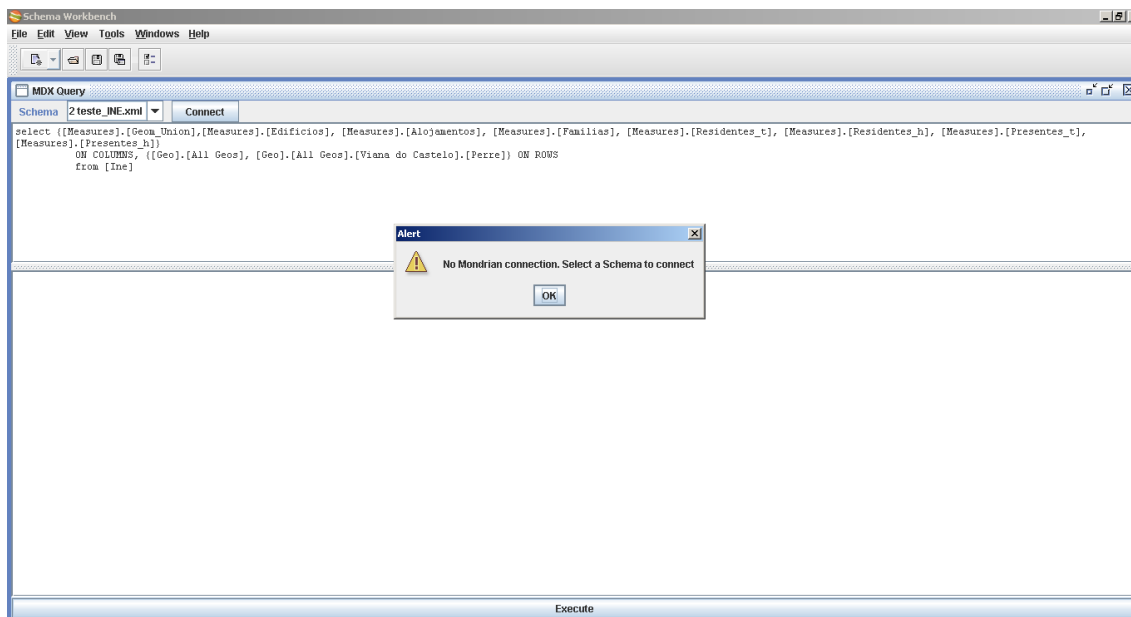


Figura 40 - Ilustração de um erro da execução de uma pesquisa MDX.

Expõe-se agora as etapas para a construção do modelo que suporta o Cubo, para a análise dos dados. A primeira acção realizada foi a criação de uma conexão entre a aplicação e o DW, onde se encontravam dos dados. A figura 41 mostra quais os parâmetros necessários para a criação da mesma.

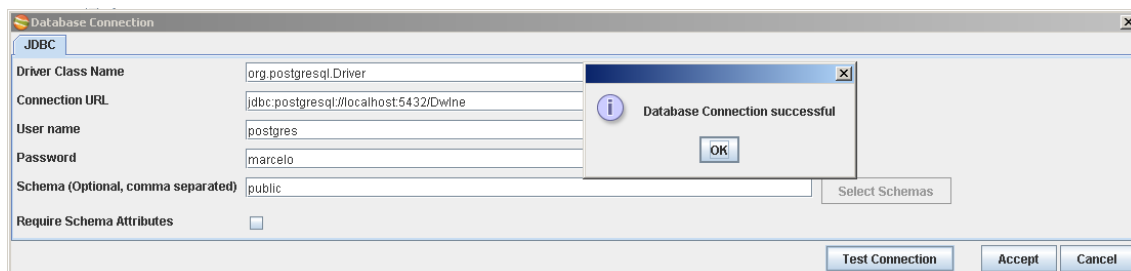


Figura 41 - Exemplificação da ligação entre o Workbench e o DW.

O Schema pode ser visualizado nas figuras 42 e 43 é constituído por:

- Cubo (ine)
- Dimensão (Geo)

- Hierarquia (default)
- Metricas (Edifícios, Alojamentos, etc)

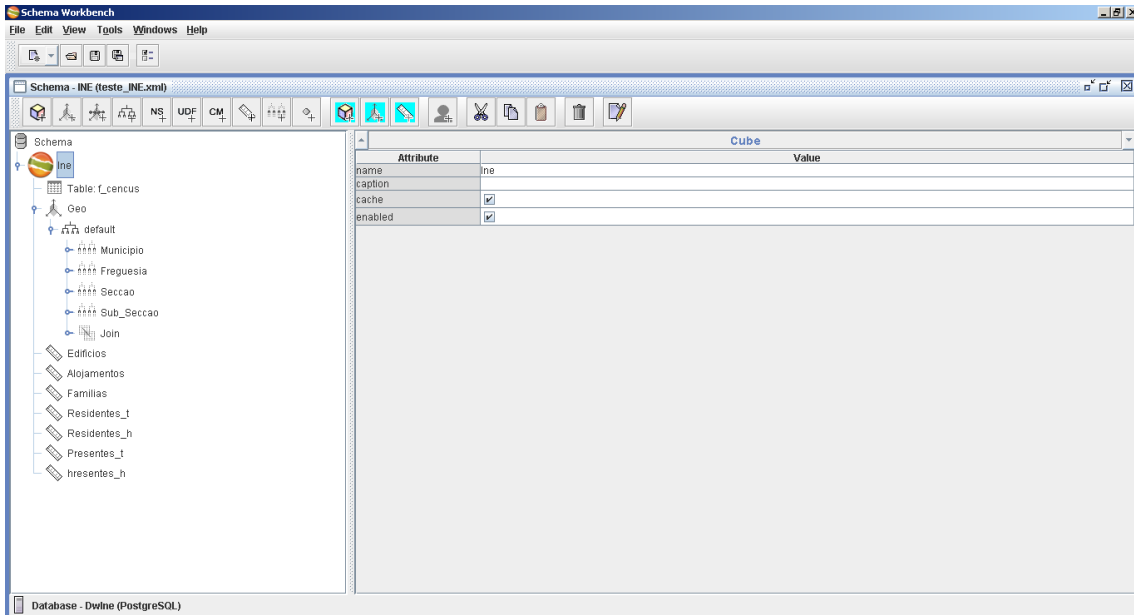


Figura 42 - Modelo da Estrutura do Cubo

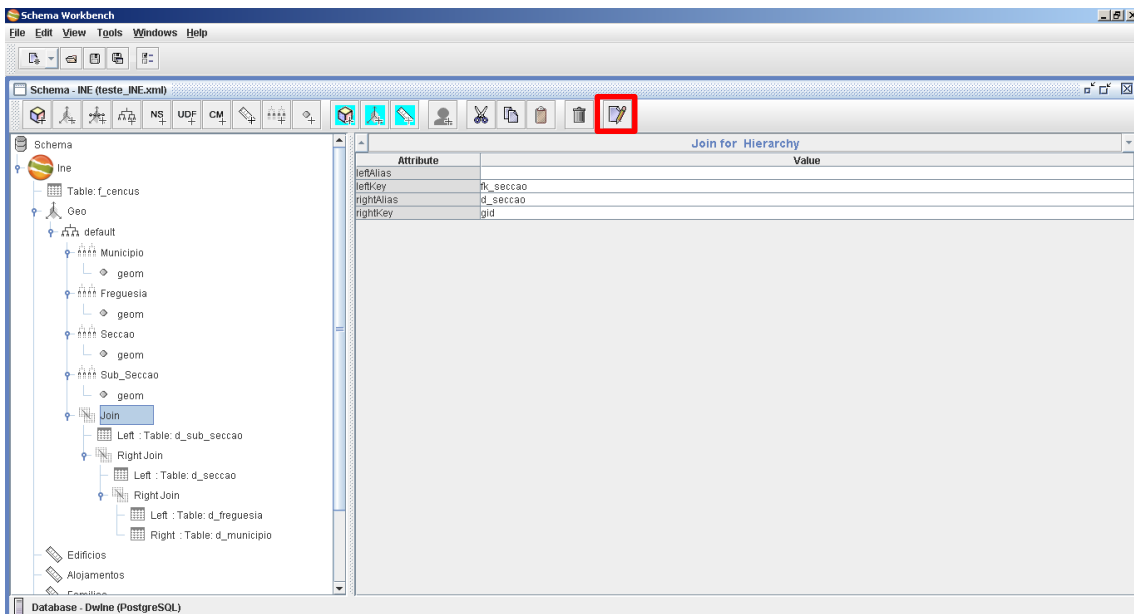


Figura 43 - Detalhe da construção do Cubo

O schema é nada mais do que um ficheiro XML, com a estrutura acima apresentada. A ferramenta permite também a visualização nesse formato, para isso é necessário clicar no botão assinalado com um rectângulo vermelho na figura 43

Na tabela 3, é possível encontrar o esquema XML do schema aqui apresentado na figura 43.

```

<?xml version="1.0"?>
<Schema name="INE">
<Cube name="Ine" >
  <Table name="f_cencus" />
  <!-- spatial dimension -->
  <Dimension name="Geo" foreignKey="fk_sub_sec">

    <Hierarchy hasAll="true" primaryKey="gid" primaryKeyTable="d_sub_seccao">
      <Join leftKey="fk_seccao" rightKey="gid" rightAlias="d_seccao">
        <Table name="d_sub_seccao" />
        <Join leftKey="fk_freguesia" rightKey="gid" rightAlias="d_freguesia">
          <Table name="d_seccao" />
          <Join leftKey="fk_municip" rightKey="gid">
            <Table name="d_freguesia" />
            <Table name="d_municipio" />
          </Join>
        </Join>
      </Join>
    </Hierarchy>
    <Level name="Municipio" table="d_municipio" column="nome" uniqueMembers="true">
      <Property name="geom" column="the_geom" type="Geometry" />
    </Level>
    <Level name="Freguesia" table="d_freguesia" column="nome" uniqueMembers="true">
      <Property name="geom" column="the_geom" type="Geometry" />
    </Level>
    <Level name="Seccao" table="d_seccao" column="seccao" uniqueMembers="true">
      <Property name="geom" column="the_geom" type="Geometry" />
    </Level>
    <Level name="Sub_Seccao" table="d_sub_seccao" column="subseccao" uniqueMembers="true">
      <Property name="geom" column="the_geom" type="Geometry" />
    </Level>
  </Dimension>
  <Measure name="Edificios" column="edificios" aggregator="sum"
formatString="Standard"></Measure>
  <Measure name="Alojamentos" column="alojamentos" aggregator="sum" formatString="#,###.00">
</Measure>
  <Measure name="Familias" column="familias" aggregator="sum"
formatString="#,###.00"></Measure>
  <Measure name="Residentes_t" column="residentes_t" aggregator="sum"
formatString="#,###.00"></Measure>
  <Measure name="Residentes_h" column="residentes_h" aggregator="sum"
formatString="#,###.00"></Measure>
  <Measure name="Presentes_t" column="presentes_t" aggregator="sum"
formatString="#,###.00"></Measure>
  <Measure name="hresentes_h" column="presentes_h" aggregator="sum"
formatString="#,###.00"></Measure>

</Cube>
</Schema>

```

Tabela 3 – Esquema XML do cubo

## 4.4.5 Visualização Geográfica

Após a conclusão das etapas anteriores é então possível apresentar os dados sobre a forma de mapa, usando as ferramentas anteriormente mencionadas. A disponibilização dos dados para o utilizador, foi desenvolvido a partir do exemplo que a ferramenta GeoMondrian oferecia. Este exemplo disponibilizava somente a visualização dos dados através de tabelas, podendo executar as acções de Drill-down e Roll-Up. Para complementar esta solução desenvolveu-se a visualização dos dados através de um mapa.

Em primeiro lugar efectuou-se a publicação do Schema construído no ponto anterior, para a pasta do projecto que se encontra publicado no servidor Apache/Tomcat, como pode ver identificado pelo rectângulo vermelho na figura 44. Seguindo as orientações do exemplo do GeoMondrian esta publicação é efectuada no seguinte *path*: “C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\PASTA DO PROJECTO\WEB-INF\queries”.

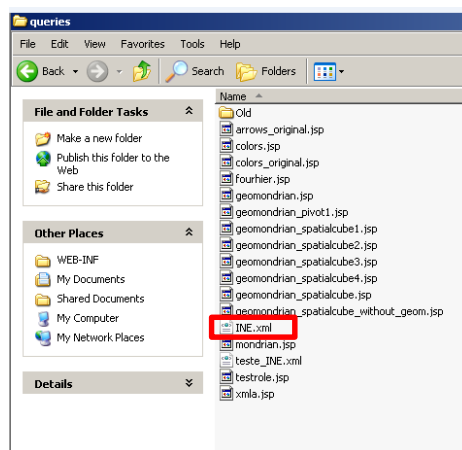


Figura 44 - Publicação do esquema no projecto.

De seguida efectuou-se as alterações necessárias no ficheiro “**Index.html**” de modo a que o objectivo fosse alcançado. A aplicação foi alterada para que o seu aspecto fica-se como ilustrado a figura 45

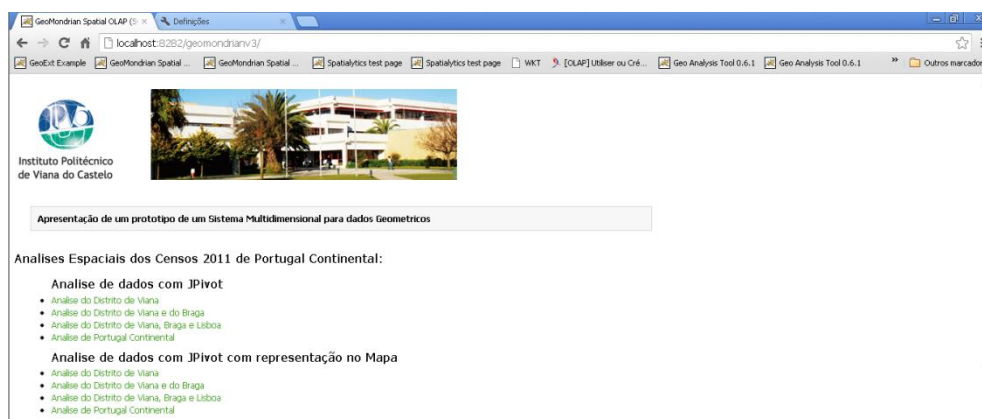


Figura 45 - Ilustração do layout da aplicação.

Como se pode verificar existem quatro links, sendo que entre estes diferem somente no conjunto de dados que tem disponível. Esta foi uma solução para avaliar o desempenho da ferramenta. As etapas para a construção da aplicação foram as seguintes:

- Alteração do ficheiro “Index.html”

Aqui os links são definidos e redireccionados para ler uma pesquisa (escrita em MDX) que limita os dados tal como a tabela 4 demonstra.

```
<li><a href="testpage.jsp?query=geomondrian_spatialcube1">Análise do Distrito de Viana</a></li>
<li><a href="testpage.jsp?query=geomondrian_spatialcube2">Análise do Distrito de Viana e do
Braga</a></li>
<li><a href="testpage.jsp?query=geomondrian_spatialcube3">Análise do Distrito de Viana, Braga e
Lisboa</a></li>
<li><a href="testpage.jsp?query=geomondrian_spatialcube4">Análise de Portugal
Continental</a></li>
```

Tabela 4 - Definição dos links para análise dos dados.

- Alteração do ficheiro “testpage.jsp”

Este ficheiro é o responsável por ler a pesquisa definida nos ficheiros “geomondrian\_spatialcube1”. A partir desta pesquisa constrói uma tabela que apresenta em forma pivot table ou no mapa. O código responsável pela atribuição da pesquisa ao parâmetro “query” é identificado na tabela 5.

```
<%-- include query and title, so this jsp may be used with different queries --%>
<wcf:include id="include01" httpParam="query" prefix="/WEB-INF/queries/" suffix=".jsp"/>
<c:if test="{query01 == null}">
  <jsp:forward page="."/>
</c:if>
```

Tabela 5 - Atribuição da pesquisa MDX ao parâmetro.

A tabela 6 mostra a secção de código que desenha a tabela onde se encontram os dados. Esta é construída dinamicamente para que todas as análises sejam possíveis.

```
<!-- render the table -->
<p>
<wcf:render ref="table01" xslUri="/WEB-INF/jpivot/table/mdxtable.xsl" xslCache="true"/>
</p>
```

Tabela 6 - Renderizar a tabela com os dados.

Este ficheiro foi ainda editado com código para que fosse possível a visualização dos dados no mapa. Este código, identificado no anexo A2, foi desenvolvido em JavaScript, sendo executado totalmente do “lado” do cliente.

Para que os dados sejam representados no mapa, é necessário preencher o Array Dados, sendo este processo efectuado neste secção de código identificada na tabela 7.

```
<script language ="JavaScript">
var linha;
for (linha=3; linha<document.getElementsByTagName("table")[1].rows.length; linha++)
{
dados.push({ 'value':
parseFloat(document.getElementsByTagName("table")[1].rows[linha].cells[5].childNodes[0].data.replace('.', '')),
'geometry':
document.getElementsByTagName("table")[1].rows[linha].cells[1].childNodes[1].data
});
}
</script>
```

Tabela 7 – Código necessário ao carregamento do Array para a visualização no mapa.

Quer o ficheiro “testpage.jsp” quer o “index.html”, encontram-se na raiz da pasta do projecto, no servidor onde esta publicado.

- Alteração do ficheiro geomondrian\_spatialcube:

Tal como já foi referido este ficheiro contém a pesquisa que delimita o conjunto de dados. O código contido na tabela 8 exemplifica o caso de Viana do Castelo.



```
<%@ 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/DwIne?user=postgres&password=marcelo"
catalogUri="/WEB-INF/queries/teste_INE.xml">
  Select {[Measures].[Edifícios], [Measures].[Alojamentos], [Measures].[Famílias],
[Measures].[Residentes_t], [Measures].[Residentes_h], [Measures].[Presentes_t],
[Measures].[Presentes_h]} on columns,
  {[Geo].[All Geos].[Viana do Castelo]} ON rows
from Ine
</jp:mondrianQuery>

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

Tabela 8 – Código da pesquisa MDX para limitar os dados ao conselho de Viana do Castelo

Tendo todas estas alterações efectuadas é possível a visualização dos dados em formato de tabela e nos mapas. A figura 46, ilustra a visualização dos dados para os quatro sub conjunto de dados, em formato de tabela sem efectuar qualquer acção de drill-down/roll-up sobre os dados.

	Measures						
Geo	Edifícios	Alojamentos	Famílias	Residentes_t	Residentes_h	Presentes_t	Presentes_h
+Viana do Castelo	32.925	48.290,00	31.767,00	88.725,00	41.895,00	85.312,00	39.698,00

	Measures						
Geo	Edifícios	Alojamentos	Famílias	Residentes_t	Residentes_h	Presentes_t	Presentes_h
+Viana do Castelo	32.925	48.290,00	31.767,00	88.725,00	41.895,00	85.312,00	39.698,00
+Braga	38.881	84.675,00	64.099,00	181.474,00	86.701,00	177.840,00	83.911,00

	Measures						
Geo	Edifícios	Alojamentos	Famílias	Residentes_t	Residentes_h	Presentes_t	Presentes_h
+Viana do Castelo	32.925	48.290,00	31.767,00	88.725,00	41.895,00	85.312,00	39.698,00
+Porto	44.314	137.879,00	101.137,00	237.584,00	108.131,00	242.232,00	109.923,00
+Lisboa	52.485	323.937,00	244.354,00	547.631,00	250.930,00	547.265,00	250.920,00

	Measures						
Geo	Edifícios	Alojamentos	Famílias	Residentes_t	Residentes_h	Presentes_t	Presentes_h
+All Geos	3.352.946	5.638.632,00	3.874.185,00	10.047.291,00	4.799.720,00	9.769.284,00	4.623.044,00

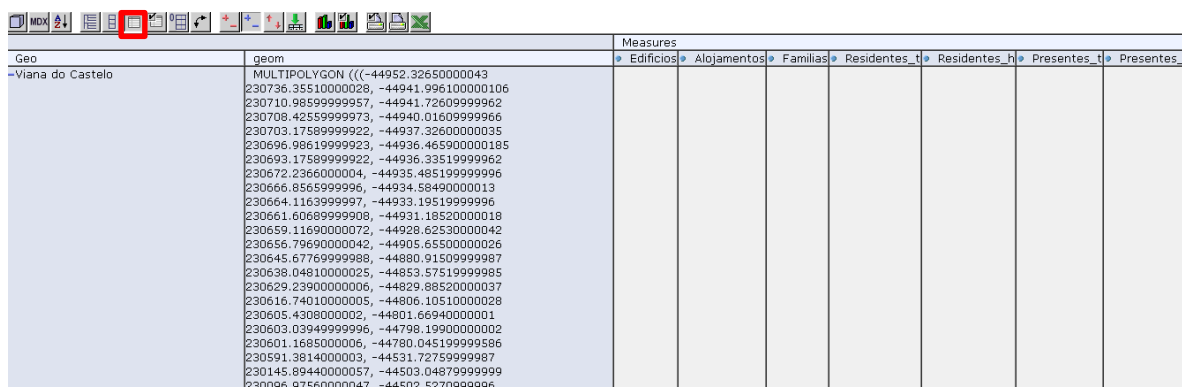
Figura 46 - Apresentação dos dados alfanuméricos.

Obtida a tabela, esta permite a realização das acções de Drill-Down e Roll-Up. Na figura 47 é demonstra a realização da acção de Drill-down sobre os dados, chegando ao nível mais baixo (subsecção).

Geo	Measures						
	Edifícios	Alojamentos	Familias	Residentes_t	Residentes_h	Presentes_t	Presentes_h
-Viana do Castelo	32.925	48.290,00	31.767,00	88.725,00	41.895,00	85.312,00	39.698,00
+Affe	1.058	1.269,00	638,00	1.632,00	749,00	1.543,00	689,00
+Alvarães	1.299	1.347,00	914,00	2.623,00	1.275,00	2.574,00	1.229,00
+Amonde	136	138,00	100,00	293,00	122,00	274,00	110,00
+Anha	1.109	1.193,00	854,00	2.415,00	1.136,00	2.317,00	1.072,00
+Areosa	1.837	2.516,00	1.760,00	4.853,00	2.268,00	4.622,00	2.134,00
+Barroselas	1.660	1.854,00	1.373,00	3.927,00	1.875,00	3.745,00	1.779,00
+Cardielos	519	530,00	396,00	1.309,00	623,00	1.224,00	569,00
-Perre	1.080	1.117,00	928,00	2.956,00	1.417,00	2.784,00	1.305,00
-160926001	393	402,00	321,00	995,00	464,00	948,00	432,00
16092600101	9	9,00	7,00	19,00	8,00	14,00	6,00
16092600102	6	7,00	6,00	19,00	7,00	18,00	6,00
16092600103	12	12,00	10,00	24,00	10,00	22,00	9,00
16092600104	5	6,00	6,00	18,00	7,00	14,00	5,00
16092600105	13	13,00	12,00	36,00	17,00	36,00	17,00
16092600106	10	10,00	7,00	20,00	10,00	18,00	8,00
16092600107	8	8,00	6,00	16,00	6,00	16,00	6,00
16092600108	34	34,00	25,00	66,00	27,00	66,00	27,00
16092600109	5	5,00	4,00	12,00	5,00	12,00	5,00
16092600110	12	13,00	12,00	32,00	16,00	33,00	17,00

Figura 47 - Ilustração das operações de Drill-down Roll-Up.

A partir deste componente é também possível a visualização dos dados geográficos, sendo para isso necessário clicar no botão identificado com o quadrado vermelho na figura 48.



Geo	geom	Measures						
		Edifícios	Alojamentos	Familias	Residentes_t	Residentes_h	Presentes_t	Presentes_h
-Viana do Castelo	MULTIPOLYGON (((-44952.32650000043 230736.35510000028, -44941.996100000106 230710.98599999957, -44941.72609999962 230708.42559999973, -44940.01609999966 230703.17589999922, -44937.32600000035 230696.98619999923, -44936.465900000185 230693.17589999922, -44936.33519999962 230672.2366000004, -44935.48519999996 230666.8565999996, -44934.58490000013 230664.1163999997, -44933.19519999996 230661.60689999908, -44931.18520000018 230659.11690000072, -44928.62530000042 230656.79690000042, -44905.65500000026 230645.67769999988, -44880.91509999987 230638.04810000025, -44853.57519999985 230629.23900000006, -44829.88520000037 230616.74010000005, -44806.10510000028 230605.4308000002, -44801.66940000001 230603.03949999996, -44798.19900000002 230601.16850000006, -44780.045199999586 230591.38140000003, -44531.72759999987 230145.89440000057, -44503.04879999999 230096.97560000047, -44502.52709999996							

Figura 48 - Apresentação de dados geométricos na Pivot table.

O resultado da selecção desta opção é da visualização dos dados sobre a forma de mapa de acordo com uma legenda predefinida. As figuras 50, 51 e 52 demonstram as acções de drill-down sobre os dados no concelho de Viana do Castelo, podendo estas operações ser esquematizadas da seguinte forma (figura 49):

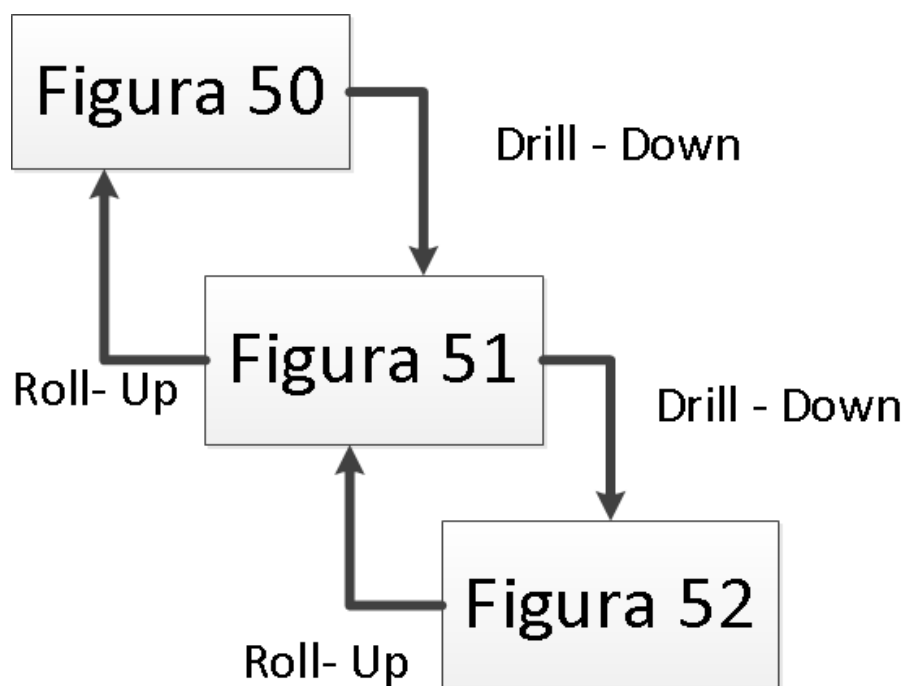


Figura 49 - Esquemático das operações Drill-Down e Roll-up sobre os dados

A figura inicial (figura 50) mostra todas as freguesias do concelho, a seguinte (figura 51) mostra as secções de uma das freguesias (neste caso, Perre) e por fim são apresentadas as subsecções de uma secção na freguesia de Perre figura 52).

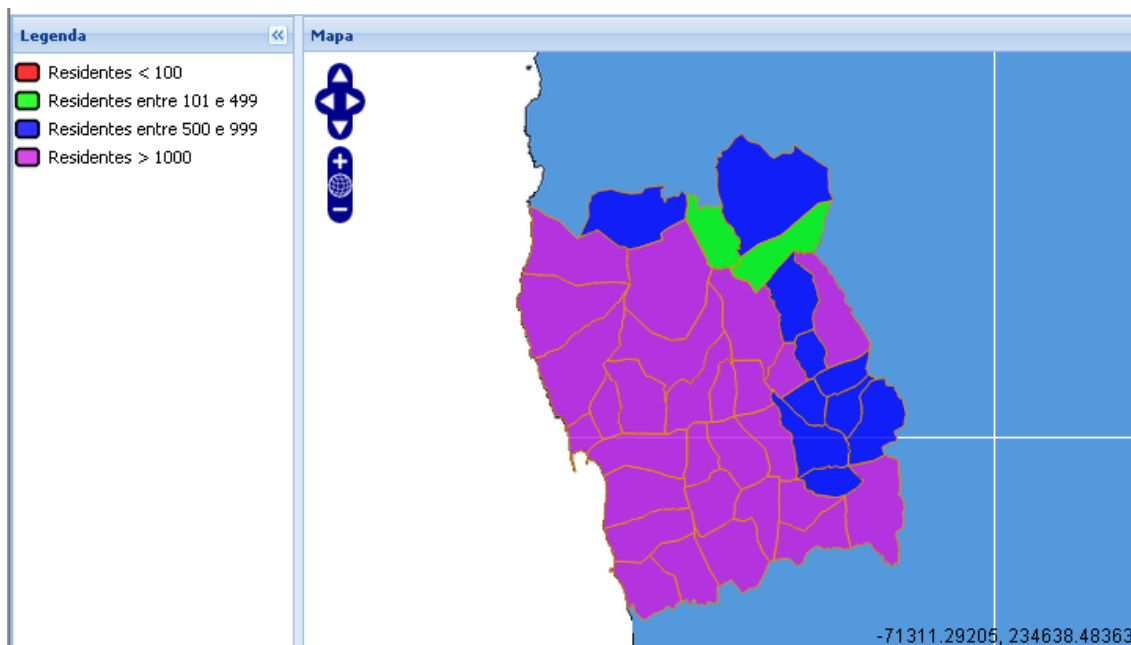


Figura 50 - Visualização ao nível das freguesias no concelho de Viana do Castelo

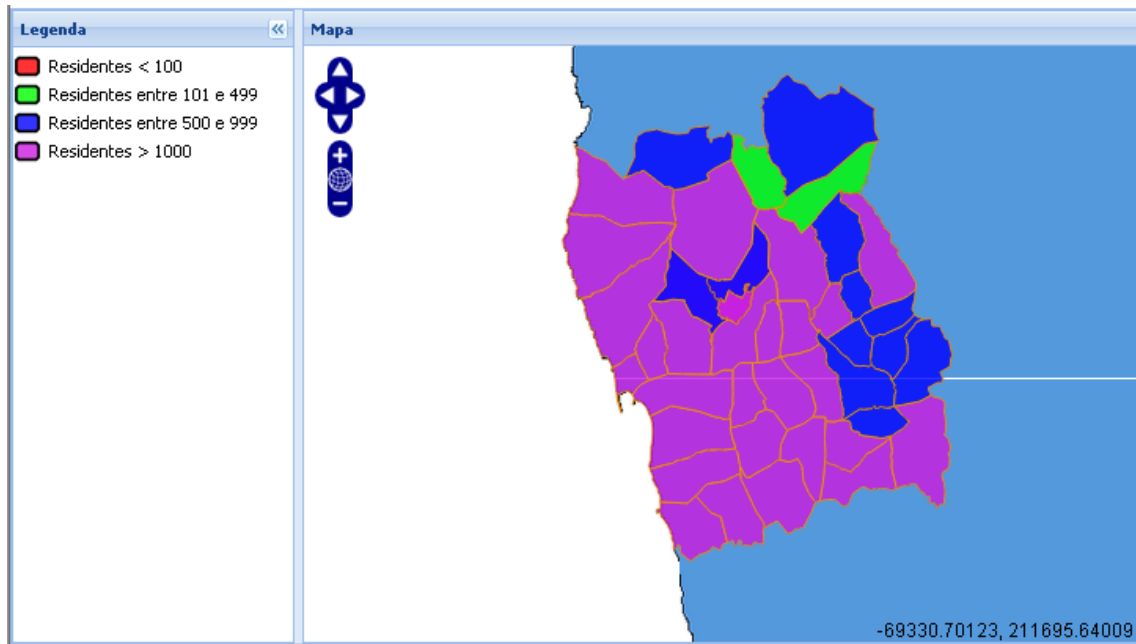


Figura 51 - Visualização ao nível das das secções da Freguesia de Perre

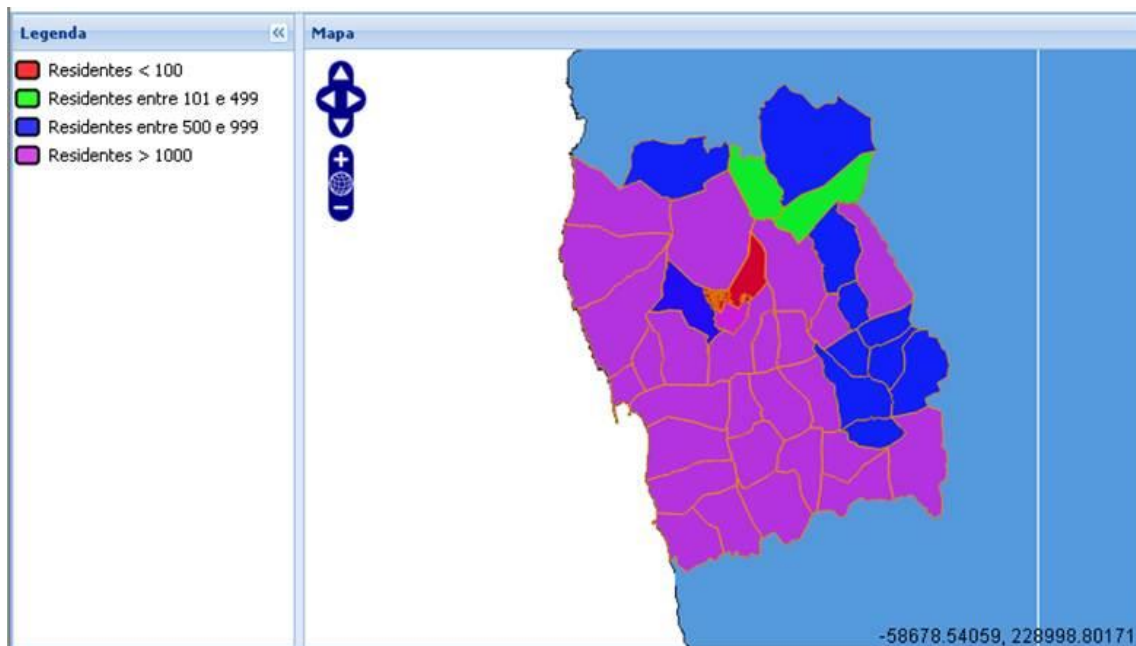


Figura 52 - Visualização ao nível das das secções da Freguesia de Perre

A figura 53 ilustra o resultado uma acção de Drill-Down sobre todo o território do país, sendo possível a visualização de todas as freguesias.

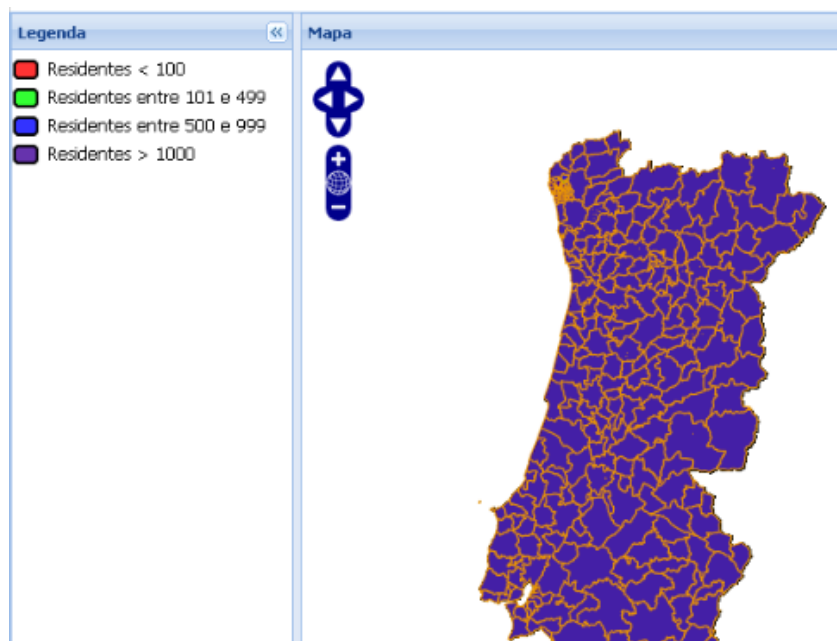


Figura 53 - Visualização geral do país.

Como um dos pontos deste projecto centrava-se na análise de desempenho das ferramentas, realizaram-se várias acções para verificar os procedimentos a adoptar para otimizar o desempenho da aplicação. Estas acções centraram-se na geração de índices no Data Warehouse, com as boas práticas o recomendam, o aumento da memória RAM disponível, a optimização dos parâmetros da base de dados, bem como testes com o servidor e cliente em máquinas diferentes.

Como primeiro teste de desempenho, efectuou-se a alteração dos valores de memória configurados no servidor web, tendo-se passado de uma memória inicial de 256Mb para 512Mb com um máximo de 1024Mb.

A alteração destes parâmetros foi notada de forma considerável no desempenho da aplicação. Tentou-se ainda aumentar para 1024Mb a configuração de memória inicial contudo este teste foi inviabilizado devido a falta de capacidade do equipamento de teste. De notar que o ambiente no qual o teste foi realizado continha 3Gb de memória RAM.

Para verificar o desempenho do processador e a memória que é utilizada no executar a aplicação, efectuaram-se diversos testes, todos seguindo a política de: o servidor ser reiniciado para não conter dados em cache, sendo a cache do browser limpa também.

Os casos de teste começaram-se por ser realizados com a adição de dados de forma incremental. Esta adição de dados, tem uma forte influência no desempenho da

aplicação. Quanto maior o volume de dados maior é a utilização do processador e de memória para alcançar os resultados pretendidos, bem como o tempo necessário para ficar disponível.

Ao longo das figuras 54, 55, 56 e 57 é possível verificar a alteração de consumos quer de processador quer de memória para diferentes volumes de dados como é o caso do Município de Viana do Castelo, dos Municípios de Viana do Castelo e Braga, dos Municípios de Viana do Castelo Braga e Lisboa e por último sobre Portugal Continental.

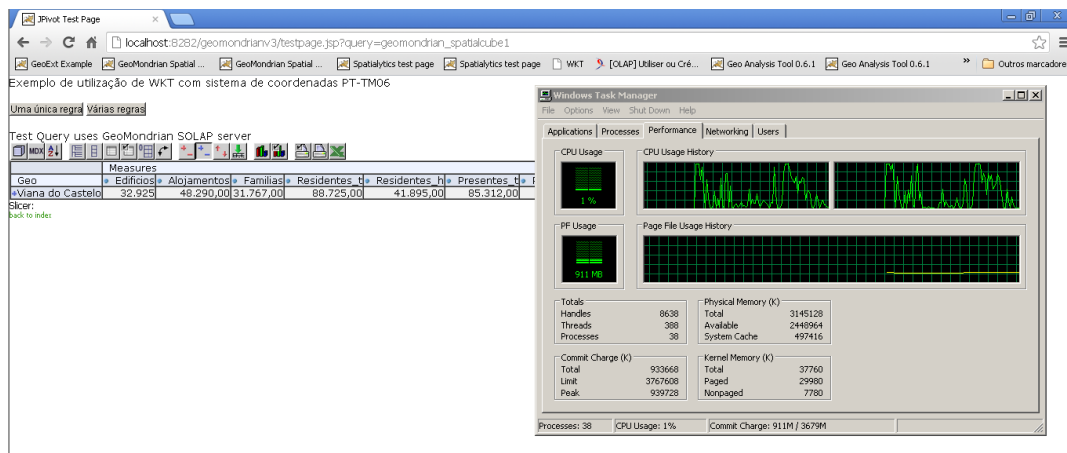


Figura 54 - Análise do desempenho do Sistema – Análise para o Município de Viana do Castelo.

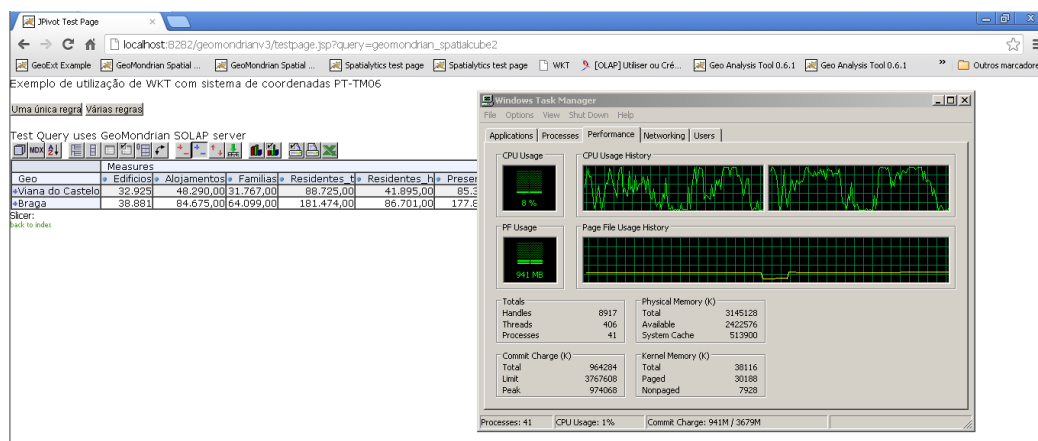


Figura 55 - Análise do desempenho do Sistema – Análise para o Município de Viana do Castelo e de Braga.

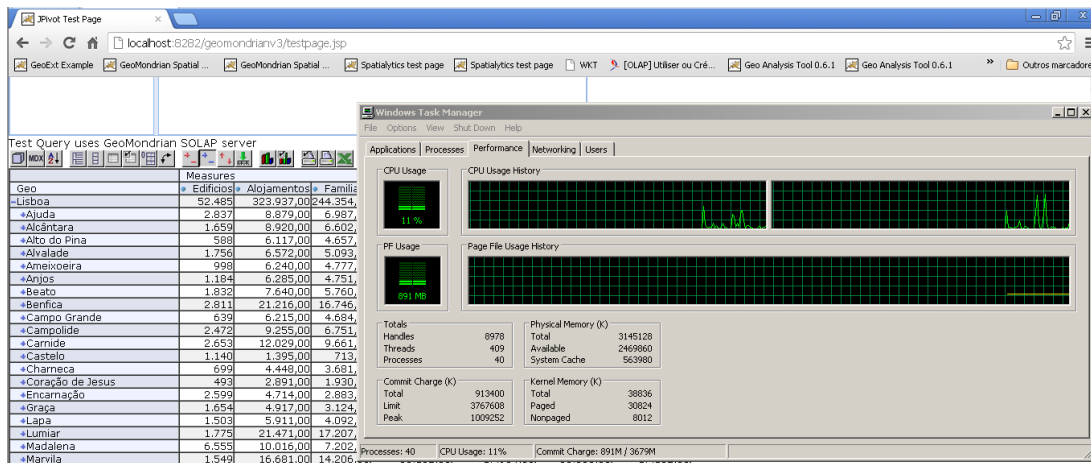


Figura 56 - Análise do desempenho do Sistema – Análise para o Município de Viana do Castelo, Braga e Lisboa.

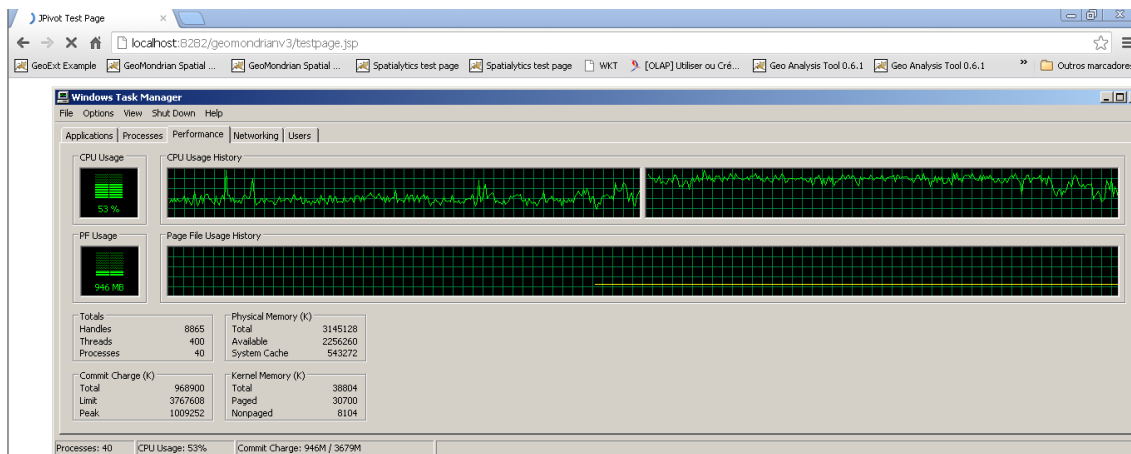


Figura 57 - Análise do desempenho do Sistema – Análise para Portugal Continental.

Por outro lado a ferramenta PGAdminIII, oferece uma interface gráfica para a gestão das bases de dados PostgreSQL, disponibiliza uma funcionalidade que permite a análise das sessões que estão activas (que actualmente estão a processar dados). Aqui é possível ver qual o tempo da pesquisa necessário para executar as consultas do GeoMondrian. O exemplo da figura 58 mostra o tempo de execução que este executou a executar, para mostrar todos os Municípios de Portugal Continental.

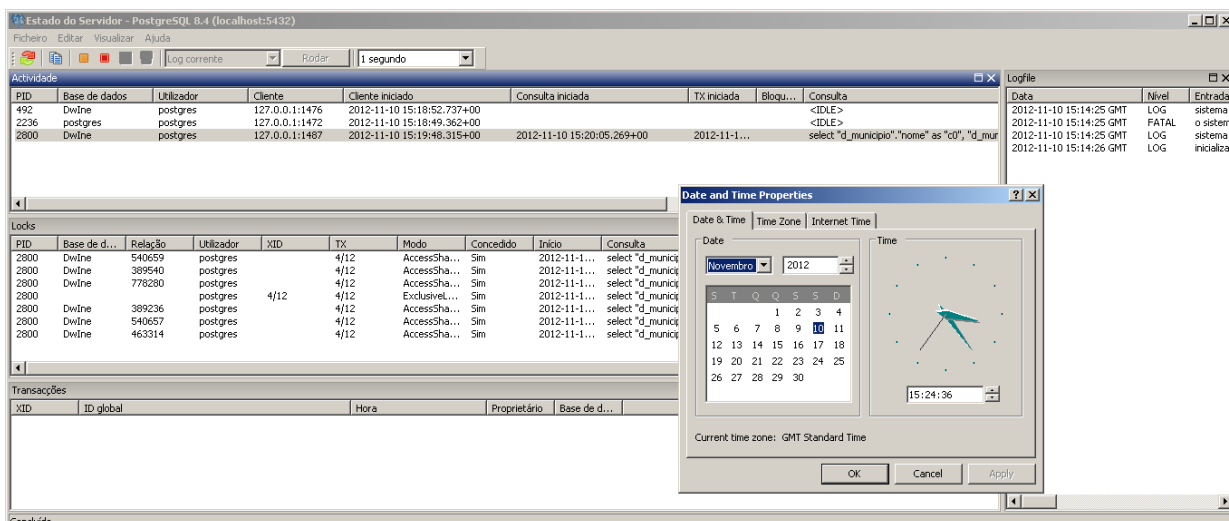


Figura 58 - Análise das sessões activas na base dados.

Este tipo de análise pode também ser efectuado pela seguinte pesquisa SQL apresentada na tabela 9.

```
select * from pg_stat_Activity;
```

Tabela 9 – Pesquisa SQL das sessões activas na base de dados.

Na tabela 10 é possível visualizar o aumento do tempo de processamento, a medida que o conjunto de dados envolvido aumenta.

Conjunto de dados	Tempo
Viana do Castelo	15 segundos
Viana do Castelo e Braga	22 segundos
Viana do Castelo, Braga e Lisboa	35 segundos
Portugal Continental	6 minutos e 8 segundos

Tabela 10 - Comparação de tempos

Estes testes foram realizados com as configurações default que o PostgreSQL gera na sua instalação. No entanto a solução disponibiliza um *plug-in* que permite a alteração destas configurações mediante o tipo de ambiente pretendido. Essa instalação foi efectuada a partir do Stack Builder, na qual se escolheu a opção identificada pelo rectângulo vermelho na figura 59.



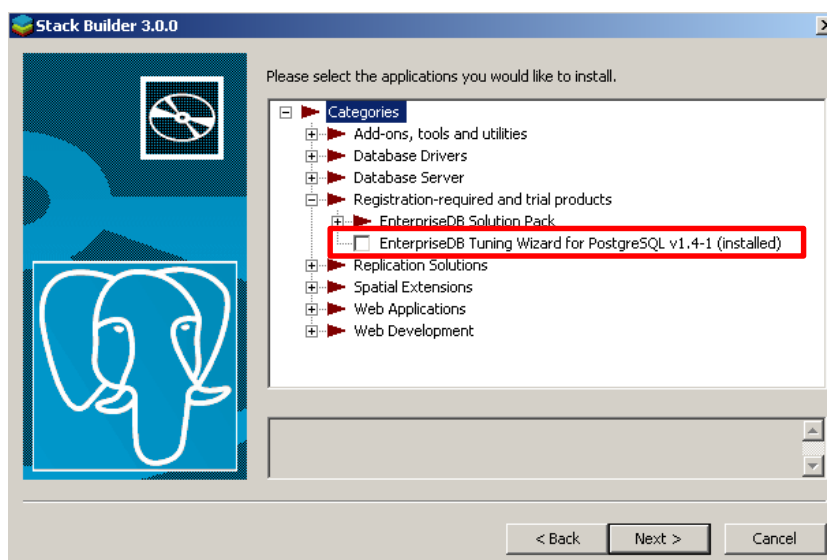


Figura 59 - Instalação do plug-in para a configuração dos parâmetros do PostgreSQL

Este *plug-in*, permite a alteração do ficheiro de configuração disponibilizando três opções:

- Ambiente de desenvolvimento
  - Tal como a ferramenta indica, não deverão ser efectuados testes de desempenho com esta configuração, pois o PostgreSQL irá usar a memória mínima.
- Ambiente com várias aplicações
  - Esta configuração deverá usada em ambientes com várias aplicações. Deverá ser também escolhida se for um Web Application Server.
- Ambiente dedicado
  - Deverá ser utilizado quando o ambiente é totalmente dedicado, para que a os recursos tenham o máximo de optimização.

No anexo A encontram-se as quatro versões do ficheiro de configuração. Todas as alterações efectuadas pelo *plug-in* estão identificadas.

Para um maior rigor nos teste de desempenho realizados, configurou-se uma máquina totalmente dedicada ao projecto, como o sistema operativo Linux, (Ubuntu Server 12.04). Neste ambiente foi configurado o PostgreSQL bem como o Apache Tomcat. As características desta máquina são substancialmente melhores que as

virtualizadas para a máquina virtual. Esta possui um processador Dual Core com 4 núcleos, 4Gb de memória e um disco dedicado,

Por sua vez, os dados foram copiados da máquina virtual utilizada para a realização do projecto através de um Dump da base de dados. Quer o Dump dos dados quer a pasta com a aplicação foram publicados neste servidor. No entanto tal como a figura 60 ilustra o uso de processador também é elevado, tal como o rectângulo a vermelho indica.

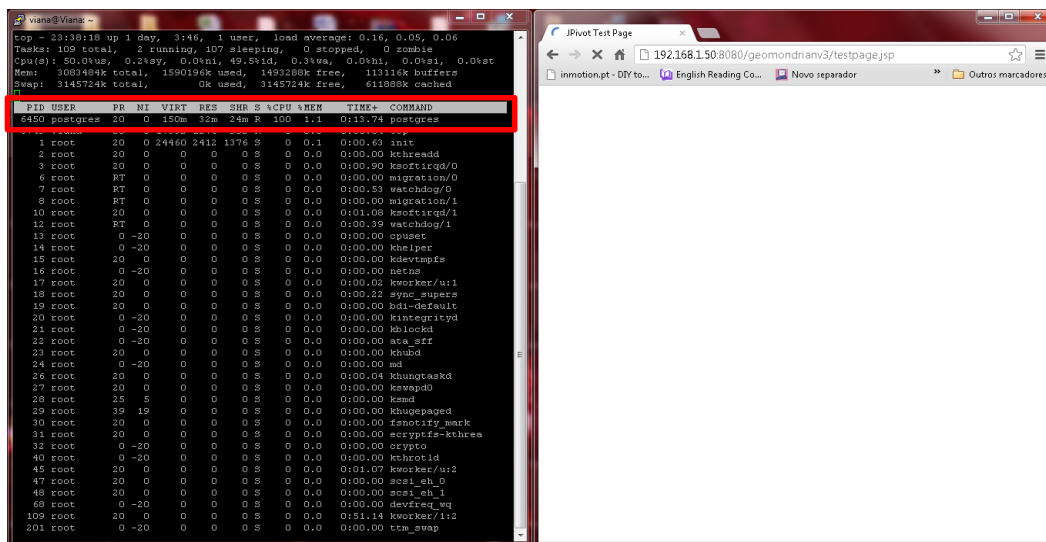


Figura 60 - Análise do desempenho do Sistema – Análise para Portugal Continental.

Apesar de o consumo quer de processador quer de memória por momento ser elevado, verificou-se que o desempenho das ferramentas aumentou com a utilização deste ambiente. Todos os testes realizados, foram efectuados sempre depois de limpar a cache da base de dados e do Apache e do browser. Este é um ponto importante pois, após o primeiro pedido ter terminado, se,for efectuado o mesmo logo de seguida, ou outro que envolva um conjunto de dados mais pequeno o tempo necessário reduz consideravelmente.

## 4.5 Análise e Discussão

Ao longo da realização do caso apresentado surgiram vários problemas com solução encontrada e outros sem resolução. Um das tarefas que se realizou tal como descrita foi a alteração do ficheiro “testpage.jsp” e geraram-se duas versões do mesmo,

para obter casos diferentes. Este passo veio a comprovar-se que não era possível e que a solução funciona exclusivamente este nome específico.

Outro problema identificado relaciona-se com a memória insuficiente pra correr a aplicação, transmitida através de um erro de execução. Analisando o estado da máquina, verifica-se que quer o processador, quer a memória RAM estão dentro dos parâmetros normais, e o próprio servidor onde a aplicação está publicada, também não apresenta qualquer problema. Não identifiquei nenhum padrão para o acontecimento desta situação pois se efectuasse o mesmo exemplo novamente este funcionava perfeitamente, sendo também que este teste foi realizado depois de reiniciar o servidor Apache e limpo a cache do browser.

A utilização do *plug-in* que permite a alteração dos parâmetros por si não garante que o sistema fique melhor do que com as condições por omissão. Estas melhorias são personalizadas/customizadas a cada ambiente, sendo necessário por vezes a sua alteração manualmente.

Por falta de tempo, o protótipo deveria ser também construído utilizando outras abordagens associadas a modelação multidimensional e comparar os resultados obtidos. Outro ponto fulcrar, seria uma optimização do processo de ETL para que este tivesse o máximo de desempenho possível. Aqui deveria-se ter em conta a manutenção incremental dos dados no DW. Este ponto é muito relevante pois tal afecta o desempenho, quer do carregamento de dados quer do processamento dos cubos. Sendo este um DW com informação geográfica é necessário sempre ponderar que uma alteração de dados a este nível pode ter como necessidade o recalculamento de toda a informação, o que pode ser um processo demorado. No entanto estes processos devem ter sempre em conta o que é mais benéfico para a organização onde estão inseridos. Tal como foi referido anteriormente, foram efectuados conjuntos de dados para que o utilizador os podesse analisar com mais rapidez. No caso de existir uma necessidade de analisar o total volume de dados, terá sempre de se chegar a um consenso com os responsáveis de negócio, pois algumas análises podem demorar algum tempo, o que por vezes para colmatar estas situações é necessário um investimento em hardware.



## **5. Conclusões**

### **5.1 Introdução**

Cada vez mais, os agentes de decisão necessitam que lhes sejam disponibilizadas ferramentas analíticas que os auxiliem no seu processo de tomada de decisão. Tradicionalmente, os Sistemas de Data Warehousing e os Sistemas de Processamento Analítico apresentam inúmeras vantagens quando se pretende analisar informação com algumas particularidades, como orientada ao assunto, informação histórica, armazenamento e processamento de grandes volumes de dados. Com as características associada à potencialidade da informação dos Sistemas de Informação Geográfica, tipicamente armazenando grandes volumes de dados, nas últimas décadas tem vindo a constatar-se uma complementaridade entre este tipo de sistemas na disponibilização analítica com a característica geográfica na disponibilização de informação analítica aos agentes de decisão, quer alfanumérica, quer com a característica de georreferenciação. Paralelamente, em termos de implementação deste tipo de sistemas de informação, tem vindo a constatar-se um forte crescimento no desenvolvimento deste tipo de sistemas usando a tecnologia open source. Neste contexto, este trabalho centrou-se no estudo, análise e exploração dos conceitos associados aos Sistemas de Suporte à Decisão aplicados a Sistemas de Informação Geográfica. Em particular, pretendeu-se estudar e aplicar os conceitos e terminologias associadas aos Sistemas de Data Warehousing, Sistemas de Processamento Analítico como forma de preparação da informação para os sistemas de descoberta de

conhecimento em bases de dados quando aplicados a Sistemas de Informação Geográfica.

## **5.2 Reflexão crítica**

Tal como foi apresentado no capítulo anterior o caso de estudo teve várias situações que dificultava a progressão dos trabalhos a um ritmo constante por motivos de falta de informação. Este ponto de falta de informação abrangeu grande parte das ferramentas utilizadas, o que para isso foi necessário o acesso a informação não oficial como fóruns e blogs. Foi sentida também uma forte presença de “contra-informação” principalmente acerca da ferramenta GeoMondrian. Aqui 90% da informação existente apresenta o mesmo exemplo e conhecido no meio por “GeoFoodMart”, no entanto veio a provar-se que a informação lá contida apresentava problemas, inviabilizando a sua utilização.

Outro aspecto a referir, relaciona-se com o facto de este projecto recorrer a um conjunto de ferramentas Open Source, resultados da análise efectuada verificou-se que uma das opções seleccionadas (SOLAP Layers) tinha sido descontinuado. Contudo, apesar de todas estas dificuldades foi possível a realização de um protótipo em que fosse possível a apresentação de informação (geográfica) através de um mapa. Aqui o aspecto funcional e gráfico não foi tido como prioritário, mas sim a análise de desempenho das aplicações. Relativamente aos resultados obtidos verifica-se que a introdução da componente geográfica na análise dos dados oferece uma mais valia, por permitir a visualização de dados sobre uma representação cartográfica, que permite uma leitura dos resultados com uma interpretação diferente das tabelas alfanuméricas, possibilitando por exemplo, visualizar relações espaciais da informação ou padrões de distribuição. No que se refere ao desempenho, os testes iniciais realizados evidenciam que a utilização destas ferramentas requerem recursos informáticos com elevada capacidade de processamento, que aumenta “muito” com o volume de dados a processar.

## **5.3 Trabalho Futuro**

Como trabalho futuro era desejável a implementação deste projecto com ferramentas de carácter proprietário, como por exemplo a Microsoft ou a Oracle, visto que ambas também já permitem a manipulação destes dados. Assim era possível a

comparação das versões Open Source e proprietário de modo a elaborar estudo ainda mais fundamentado acerca das ferramentas apresentadas.

Outro aspecto a ter em conta corresponde à separação do servidor de Base de Dados e o Servidor Web, assim como analisar o impacto de os dois serviços estarem a correr em paralelo, fazendo testes com vários utilizadores em simultâneo a efectuar pedidos. Esta escolha de ferramentas pagas ou livres tem cada vez mais importância, principalmente no momento económico-financeiro que o país atravessa.

Um outro ponto de trabalho futuro seria a realização de uma pesquisa mais abrangente em termos de parâmetros de configuração do GeoMondrian, utilizado como referência o Mondrian, visto que ao longo do projecto se utilizou sempre os parâmetros por omissão. Seria interessante também uma análise mais aprofundada de cada parâmetro de configuração do PostgreSQL, de modo a otimizar o desempenho de cada ambiente.





## **6. Referências Bibliográficas**

- ANDERSON, D., ERNST, C., HORVATH, E., PARHAM, T., NICHOLLS, R., LYTLE, D., WILLIAMS, K., NECHERO, S., LEGLEITHER, K., DAVIS, L. L., DANIELS, D., ORTIZ, E., YOUNG, E., SOLTYKA, N., MCMARTY, T. & FROSH, R. 2002a. Implementation of Geospatial Data Warehouses. United States Department of Agriculture.
- ANDERSON, D., PARHAM, T., NICHOLLS, R., CARLSON, J., ROBERSON, R., MCCARTY, T. & SOLTYKA, N. 2002b. Implementation of Geospatial Data Warehouses II. In: AGRICULTURE, U. S. D. O. (ed.).
- BALTZER, O. 2011. Computational Methods for Spatial OLAP. Douturamento, Dalhousie University.
- BANSAL, V. K. 2010. Application of geographic information systems in construction safty planning. International Journal of Project Management, 66-77.
- BÉDARD, Y., MERRETT, T. & HAN, J. 2001. Fundamentals of spatial data warehousing for geographic knowledge discovery. Geographic Data MIning and Knowledge Discovery.
- BERSON & SMITH. 1997. Data Warehousing, Data Mining, & OLAP, McGraw-Hill, 1997.
- BIMONTE, S. 2010. On Modeling and analysis of Multidimensional Geographic Databases. France: LIRIS(Laboratoire d'InfoRmatique en Images et Systèmes d'information).
- BUTTENFIELD, B., GAHEGAN, M., MILLER, H. & YUAN, M. Geospatial Data Mining and Knowledge Discovery. Research Challenges in Geographic Information Science.

- BÖHM, K., MEHLER-BICHER, A. & FENCHEL, D. 2011. GeoVisualAnalytics in the Public Health Sector.
- CHEN, K., RUI, J. & WEN, J. 2011. Development of Shanghai Urban Disaster Information. System Based on Open Source WebGIS. IEEE.
- CHENG-FANG, L., TAO, M., TIE-CHENG, W., JIAN-PING, W., WEI, Y. & BI-YU, W. 2010. Research on public GIS information platform in petroleum exploration and development field.
- DARMONT, J. & BOUSSAID, O. 2006. Processing and Managing Complex Data for Decision Support, Idea Group Publishing.
- ESRI July 1998. ESRI Shapefile Technical Description. An ESRI White Paper.
- FARIA, N. A. D. S. 2006. Suporte à edição cooperativa de Informação Geográfica em ambiente Web. Mestrado, Universidade do Minho.
- FERRAZ, V. R. T. & SANTOS, M. T. P. 2010. GlobeOLAP: Improving the Geospatial Realism in Multidimensional Analysis Environment. 12th International Conference on Enterprise Information Systems.
- FERREIRA, N. C. 2006. Apostila de Sistemas de Informação Geográficas.
- FIDALGO, R. N., TIMES, V. C., SILVA, J. & SOUZA, F. F. 2004. GeoDWFrame: A Framework for Guiding the Design of Geographical Dimensional Schemas DaWaK, 26-37.
- FU, P. & SUN, J. 2010. GIS in the Web Era. Web GIS: Principles and Applications. ESRI Press.
- GRAY, J. CHAUDHURI, S. BOSWORTH, et al. 1997. Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-tables and Sub-Totals, in Data Mining and Knowledge Discovery, Vol.1, No.1, páginas: 29-53.
- GUPTA A. & MUMICK, I. 1999. Materialized Views, Techniques, Implementations, and Applications”, The MIT Press.
- INMON, W.H. 1996, Building the Data Warehouse, John Wiley & Sons.
- JENSEN, C. S., KLIGYS, A., PEDERSEN, T. B. & TIMKO, I. 2002. Multidimensional data modeling for location-based services. The VLDB Journal, 13, 1-21.
- KIMBALL, R., REEVES, L., ROSS, M., THOMTHWAIT, W., 1998. The Data Warehouse Lifecycle ToolKit – Expert Methods for Designing, Developing, and Deploying Data Warehouses, John Wiley & Sons.
- LIANBO, W., GUANGRONG, J. & HAIYONG, Z. 2010. Management Information System for Liaohe River Estuary Wetland Based on Open Source WebGIS. IEEE.

- LÓPEZ, I. F. V., SNODGRASS, R. T. & MOON, B. 2005. Spatiotemporal Aggregate Computation: A Survey. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17.
- LO, C. P. & YEUNG, A. K. Y. 2007. *Concepts and Techniques of Geographic Information Systems*, Prentice Hall.
- MARCHAND, P., BRISEBOIS, A., BÉDARD, Y. & EDWARDS, G. 2003. Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis. *ISPRS Journal of Photogrammetry & Remote Sensing*.
- MATIAS, R. & MOURA-PIRES, J. 2005. Spatial On-Line Analytical Processing (SOLAP): A Tool to analyze the Emissions of Pollutants in Industrial Installations.
- MILLER, H., HAN, J. 2009. *Geographic Data Mining and Knowledge Discovery*, second edition, Taylor & Francis Group, CRC Press.
- PEDERSEN, T. B. & TRYFONA, N. 2001. Pre-aggregation in Spatial Data Warehouses. *LNCS: 2121, Proceedings of the 7th Int. Symposium on Advances in Spatial and Temporal Databases*, 460-480.
- RIVEST, S., BÉDARD, Y. & MARCHAND, P. 2001. Toward Better Support for Spatial Decision Making: Defining the Characteristics of Spatial On-Line Analytical Processing (SOLAP). *Geomatica*.
- SHEKHAR, S., LU, C. T., TAN, T., CHAWLA, S. & VATSAVAI, R. R. 2001.
- SILVA, R. F. 2010. SOLAP+. Mestrado, Universidade Nova de Lisboa.
- SIQUEIRA, T. L. L. 2009. SB-INDEX: Um Índice Espacial Baseado em Bitmap para Data Warehouse Geográfico. Mestrado, Universidade Federal de São Carlos.
- SHEKHAR, S., LU, C. T., TAN, T., CHAWLA, S. & VATSAVAI, R. R. 2001.
- STEINIGER, S. & HUNTER, A. J. S. 2010. Free and Open Source GIS Software for Building a Spatial Data Infrastructure
- TANIAR, D. 2009. *Progressive Methods in Data Warehousing and Business Intelligence Concepts and Competitive Analytics*, Information Science Reference.
- TERESO, M. & BERNARDINO, J. 2011. *Open Source Business Intelligence Tools for SMEs*.
- TRIPATHY, A., MISHRA, L. & PATRA, P. K. 2010. A Multi Dimensional Design Framework for Querying Spatial Data using Concept Lattice. *International Advance Computing Conference*.
- [UCGIS, 2010] "Geospatial Data Mining and Knowledge Discovery", A UCGIS White Paper on Emergent Research Themes Submitted to UCGIS Research Committee by Barbara Buttenfield (University of Colorado), Mark Gahegan (Penn State

University), Harvey Miller (University of Utah), and May Yuan (University of Oklahoma).

WANG, Y. & ZHANG, M. 2010. OpenGIS - Based Lightning Information System for Electric Power System. International Conference on Intelligent System Design and Engineering Application.

YING, L., YANG, W., YA-FU, C. & LI-ZHOU, F. 2012. The Application of Spatial Data Warehouse Tecnology in The National Petroleum Resources Database. 20 10 2nd Conference on Environmental Science and Information Application Technology, 4.

ZHANG, L., LI, Y., RAO, F., YU, X., CHEN, Y. & LIU, D. An Aproach to Enabling Spatial OLAP by Aggregating on Spatial Hierarchy.

# Anexos

## A.1 – Código javascript utilizado na construção da interface Web

```
var dados = [];  
var detalhes = [];  
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'  
                }  
              }  
            ];  
var regra2 = [ { 'limsup': 100,  
                'legenda':'Residentes < 100',  
                'style': { fillColor: "#FF0000", fillOpacity: 0.8  
                }  
              },  
              { 'liminf':101, 'limsup': 499,  
                'legenda':'Residentes entre 101 e 499',  
                'style': { fillColor: "#00FF00", fillOpacity: 0.8  
                }  
              },  
              { 'liminf': 500, 'limsup': 999,  
                'legenda':'Residentes entre 500 e 999',  
                'style': { fillColor: "#0000FF", fillOpacity: 0.8  
                }  
              },  
              { 'liminf': 1000,  
                'legenda':'Residentes > 1000',  
                'style': { fillColor: "#400099", fillOpacity: 0.8  
                }  
              }  
            ];  
  
function geraRegra1() {carregaMapa(dados,regra1); }  
function geraRegra2() {carregaMapa(dados,regra2); }  
var maxBounds = new OpenLayers.Bounds(-130000,-310000,170000,280000 );  
Ext.BLANK_IMAGE_URL = "ext-3.4.0/resources/images/default/s.gif";  
var mapa;  
var lyrVector;  
  
Ext.onReady(function() {  
    // definição das opções do mapa  
    var options = {  
        controls: [new OpenLayers.Control.Navigation()],  
        maxExtent: maxBounds,  
    }  
});
```

```

        maxResolution: "auto",
        projection: "EPSG:3763",
        units: 'm'
        // ,allOverlays: true // Não existem "Base layers"
    };
    // Criar o mapa
    mapa = new OpenLayers.Map();
    mapa.setOptions(options);

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.opengis.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();
});

function carregaMapa(lista, regras) {
    // remove todos os elementos desenhados anteriormente caso existam
    lyrVector.removeAllFeatures();
    // Criar o objecto 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 objecto 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 um objecto 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 objecto 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
            }));
    }
    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;
}
```



## A.2- Ficheiro de configuração Default do ficheiro postgresql.conf

```
# -----
# PostgreSQL configuration file
# -----
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes          Time units:  ms = milliseconds
#                 MB = megabytes          s           = seconds
#                 GB = gigabytes          min          = minutes
#                                           h           = hours
#                                           d           = days
#-----
# FILE LOCATIONS
#-----
#
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.
#
#data_directory = 'ConfigDir'           # use data in another directory
#                                           # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'     # host-based authentication file
#                                           # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf'  # ident configuration file
#                                           # (change requires restart)
#
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = '(none)'           # write an extra PID file
#                                           # (change requires restart)
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
```

```
# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost', '*' = all
                                # (change requires restart)
port = 5432                     # (change requires restart)
max_connections = 100          # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = ''      # (change requires restart)
#unix_socket_group = ''         # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
                                # (change requires restart)
#bonjour_name = ''             # defaults to the computer name
                                # (change requires restart)

# - Security and Authentication -

#authentication_timeout = 1min  # 1s-600s
#ssl = off                      # (change requires restart)
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers
                                # (change requires restart)
#ssl_renegotiation_limit = 512MB # amount of data between renegotiations
#password_encryption = on
#db_user_namespace = off

# Kerberos and GSSAPI
#krb_server_keyfile = ''
#krb_srvname = 'postgres'      # (Kerberos only)
#krb_caseins_users = off

# - TCP Keepalives -
# see "man 7 tcp" for details

#tcp_keepalives_idle = 0       # TCP_KEEPIDL, in seconds;
                                # 0 selects the system default
#tcp_keepalives_interval = 0  # TCP_KEEPINTVL, in seconds;
                                # 0 selects the system default
#tcp_keepalives_count = 0     # TCP_KEEPCNT;
                                # 0 selects the system default

#-----
# RESOURCE USAGE (except WAL)
#-----
# - Memory -

shared_buffers = 32MB         # min 128kB
                                # (change requires restart)
#temp_buffers = 8MB          # min 800kB
#max_prepared_transactions = 0 # zero disables the feature
                                # (change requires restart)
# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 1MB              # min 64kB
```

```
#maintenance_work_mem = 16MB          # min 1MB
#max_stack_depth = 2MB                # min 100kB

# - Kernel Resource Usage -

#max_files_per_process = 1000        # min 25
#                                     # (change requires restart)
#shared_preload_libraries = ''       # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0ms              # 0-100 milliseconds
#vacuum_cost_page_hit = 1            # 0-10000 credits
#vacuum_cost_page_miss = 10         # 0-10000 credits
#vacuum_cost_page_dirty = 20        # 0-10000 credits
#vacuum_cost_limit = 200             # 1-10000 credits

# - Background Writer -

#bgwriter_delay = 200ms              # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100         # 0-1000 max buffers written/round
#bgwriter_lru_multiplier = 2.0      # 0-10.0 multiplier on buffers scanned/round

# - Asynchronous Behavior -

#effective_io_concurrency = 1        # 1-1000. 0 disables prefetching

#-----
# WRITE AHEAD LOG
#-----

# - Settings -

#fsync = on                          # turns forced synchronization on or off
#synchronous_commit = on             # immediate fsync at commit
#wal_sync_method = fsync              # the default is the first option
#                                     # supported by the operating system:
#                                     #   open_datasync
#                                     #   fdatasync (default on Linux)
#                                     #   fsync
#                                     #   fsync_writethrough
#                                     #   open_sync

#full_page_writes = on               # recover from partial page writes
#wal_buffers = 64kB                  # min 32kB
#                                     # (change requires restart)
#wal_writer_delay = 200ms            # 1-10000 milliseconds

#commit_delay = 0                    # range 0-100000, in microseconds
#commit_siblings = 5                 # range 1-1000

# - Checkpoints -

#checkpoint_segments = 3             # in logfile segments, min 1, 16MB each
#checkpoint_timeout = 5min           # range 30s-1h
#checkpoint_completion_target = 0.5  # checkpoint target duration, 0.0 - 1.0
#checkpoint_warning = 30s            # 0 disables
```

```
# - Archiving -

#archive_mode = off          # allows archiving to be done
                             # (change requires restart)
#archive_command = ''       # command to use to archive a logfile segment
#archive_timeout = 0        # force a logfile segment switch after this
                             # number of seconds; 0 disables

#-----
# QUERY TUNING
#-----

# - Planner Method Configuration -

#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_indexscan = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on

# - Planner Cost Constants -

#seq_page_cost = 1.0        # measured on an arbitrary scale
#random_page_cost = 4.0    # same scale as above
#cpu_tuple_cost = 0.01     # same scale as above
#cpu_index_tuple_cost = 0.005 # same scale as above
#cpu_operator_cost = 0.0025 # same scale as above
#effective_cache_size = 128MB

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
#geqo_effort = 5           # range 1-10
#geqo_pool_size = 0       # selects default based on effort
#geqo_generations = 0     # selects default based on effort
#geqo_selection_bias = 2.0 # range 1.5-2.0

# - Other Planner Options -

#default_statistics_target = 100 # range 1-10000
#constraint_exclusion = partition # on, off, or partition
#cursor_tuple_fraction = 0.1    # range 0.0-1.0
#from_collapse_limit = 8
#join_collapse_limit = 8        # 1 disables collapsing of explicit
                                 # JOIN clauses

#-----
# ERROR REPORTING AND LOGGING
#-----

# - Where to Log -
```

```
log_destination = 'stderr'          # Valid values are combinations of
                                     # stderr, csvlog, syslog and eventlog,
                                     # depending on platform.  csvlog
                                     # requires logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on              # Enable capturing of stderr and csvlog
                                     # into log files. Required to be on for
                                     # csvlogs.
                                     # (change requires restart)

# These are only used if logging_collector is on:
#log_directory = 'pg_log'          # directory where log files are written,
                                     # can be absolute or relative to PGDATA
#log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log' # log file name pattern,
                                     # can include strftime() escapes
#log_truncate_on_rotation = off    # If on, an existing log file of the
                                     # same name as the new log file will be
                                     # truncated rather than appended to.
                                     # But such truncation only occurs on
                                     # time-driven rotation, not on restarts
                                     # or size-driven rotation.  Default is
                                     # off, meaning append to existing files
                                     # in all cases.
#log_rotation_age = 1d             # Automatic rotation of logfiles will
                                     # happen after that time.  0 disables.
#log_rotation_size = 10MB         # Automatic rotation of logfiles will
                                     # happen after that much log output.
                                     # 0 disables.

# These are relevant when logging to syslog:
#syslog_facility = 'LOCAL0'
#syslog_ident = 'postgres'

#silent_mode = off                 # Run server silently.
                                     # DO NOT USE without syslog or
                                     # logging_collector
                                     # (change requires restart)

# - When to Log -

#client_min_messages = notice      # values in order of decreasing detail:
                                     #  debug5
                                     #  debug4
                                     #  debug3
                                     #  debug2
                                     #  debug1
                                     #  log
                                     #  notice
                                     #  warning
                                     #  error

#log_min_messages = warning        # values in order of decreasing detail:
                                     #  debug5
                                     #  debug4
                                     #  debug3
```

```
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic

#log_error_verbosity = default          # terse, default, or verbose messages

#log_min_error_statement = error # values in order of decreasing detail:
# debug5
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic (effectively off)

#log_min_duration_statement = -1 # -1 is disabled, 0 logs all statements
# and their durations, > 0 logs only
# statements running at least this number
# of milliseconds

# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_checkpoints = off
#log_connections = off
#log_disconnections = off
#log_duration = off
#log_hostname = off
log_line_prefix = '%t '                # special values:
# %u = user name
# %d = database name
# %r = remote host and port
# %h = remote host
# %p = process ID
# %t = timestamp without milliseconds
# %m = timestamp with milliseconds
# %i = command tag
# %c = session ID
# %l = session line number
# %s = session start timestamp
# %v = virtual transaction ID
# %x = transaction ID (0 if none)
# %q = stop here in non-session
```

```

#          processes
#  %% = '%'
# e.g. '<u%%d> '
#log_lock_waits = off          # log lock waits >= deadlock_timeout
#log_statement = 'none'       # none, ddl, mod, all
#log_temp_files = -1          # log temporary files equal or larger
                                # than the specified size in kilobytes;
                                # -1 disables, 0 logs all temp files
#log_timezone = unknown       # actually, defaults to TZ environment
                                # setting

#-----
# RUNTIME STATISTICS
#-----

# - Query/Index Statistics Collector -

#track_activities = on
#track_counts = on
#track_functions = none        # none, pl, all
#track_activity_query_size = 1024
#update_process_title = on
#stats_temp_directory = 'pg_stat_tmp'

# - Statistics Monitoring -

#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off

#-----
# AUTOVACUUM PARAMETERS
#-----
#autovacuum = on                # Enable autovacuum subprocess? 'on'
                                # requires track_counts to also be on.
#log_autovacuum_min_duration = -1 # -1 disables, 0 logs all actions and
                                # their durations, > 0 logs only
                                # actions running at least this number
                                # of milliseconds.
#autovacuum_max_workers = 3      # max number of autovacuum subprocesses
#autovacuum_naptime = 1min       # time between autovacuum runs
#autovacuum_vacuum_threshold = 50 # min number of row updates before
                                # vacuum
#autovacuum_analyze_threshold = 50 # min number of row updates before
                                # analyze
#autovacuum_vacuum_scale_factor = 0.2 # fraction of table size before vacuum
#autovacuum_analyze_scale_factor = 0.1 # fraction of table size before analyze
#autovacuum_freeze_max_age = 200000000 # maximum XID age before forced vacuum
                                # (change requires restart)
#autovacuum_vacuum_cost_delay = 20ms # default vacuum cost delay for
                                # autovacuum, in milliseconds;
                                # -1 means use vacuum_cost_delay
#autovacuum_vacuum_cost_limit = -1 # default vacuum cost limit for
                                # autovacuum, -1 means use
                                # vacuum_cost_limit

#-----
# CLIENT CONNECTION DEFAULTS
#-----

# - Statement Behavior -

```

```

#search_path = '$user',public'          # schema names
#default_tablespace = ''                # a tablespace name, '' uses the default
#temp_tablespaces = ''                 # a list of tablespace names, '' uses
                                        # only default tablespace

#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#session_replication_role = 'origin'
#statement_timeout = 0                 # in milliseconds, 0 is disabled
#vacuum_freeze_min_age = 50000000
#vacuum_freeze_table_age = 150000000
#xmlbinary = 'base64'
#xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, dmy'
#intervalstyle = 'postgres'
#timezone = unknown                    # actually, defaults to TZ environment
                                        # setting
#timezone_abbreviations = 'Default'    # Select the set of available time zone
                                        # abbreviations. Currently, there are
                                        #   Default
                                        #   Australia
                                        #   India
                                        # You can create your own file in
                                        # share/timezonesets/.

#extra_float_digits = 0                # min -15, max 2
#client_encoding = sql_ascii           # actually, defaults to database
                                        # encoding

# These settings are initialized by initdb, but they can be changed.
lc_messages = 'Portuguese, Portugal'   # locale for system error message
                                        # strings
lc_monetary = 'Portuguese, Portugal'    # locale for monetary formatting
lc_numeric = 'Portuguese, Portugal'     # locale for number formatting
lc_time = 'Portuguese, Portugal'        # locale for time formatting

# default configuration for text search
default_text_search_config = 'pg_catalog.simple'

# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''

#-----
# LOCK MANAGEMENT
#-----

#deadlock_timeout = 1s
#max_locks_per_transaction = 64        # min 10
                                        # (change requires restart)

# Note: Each lock table slot uses ~270 bytes of shared memory, and there are
# max_locks_per_transaction * (max_connections + max_prepared_transactions)
# lock table slots.

```



```
#-----  
# VERSION/PLATFORM COMPATIBILITY  
#-----  
  
# - Previous PostgreSQL Versions -  
  
#add_missing_from = off  
#array_nulls = on  
#backslash_quote = safe_encoding # on, off, or safe_encoding  
#default_with_oids = off  
#escape_string_warning = on  
#regex_flavor = advanced          # advanced, extended, or basic  
#sql_inheritance = on  
#standard_conforming_strings = off  
#synchronize_seqscans = on  
  
# - Other Platforms and Clients -  
  
#transform_null_equals = off  
  
#-----  
# CUSTOMIZED OPTIONS  
#-----  
  
#custom_variable_classes = ''      # list of custom variable class names
```

### A.3 - Ficheiro de configuração Development do ficheiro postgresql.conf

```
# -----
# PostgreSQL configuration file
# -----
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes          Time units:  ms = milliseconds
#                 MB = megabytes          s           = seconds
#                 GB = gigabytes          min          = minutes
#                                           h           = hours
#                                           d           = days
#-----
# FILE LOCATIONS
#-----
#
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.
#
#data_directory = 'ConfigDir'           # use data in another directory
#                                           # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'     # host-based authentication file
#                                           # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf'  # ident configuration file
#                                           # (change requires restart)
#
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = '(none)'           # write an extra PID file
#                                           # (change requires restart)
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
#
listen_addresses = '*'                  # what IP address(es) to listen on;
#                                           # comma-separated list of addresses;
```

```

# defaults to 'localhost', '*' = all
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = '' # (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
#bonjour_name = '' # defaults to the computer name
# (change requires restart)

# - Security and Authentication -

#authentication_timeout = 1min # 1s-600s
#ssl = off # (change requires restart)
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers
# (change requires restart)
#ssl_renegotiation_limit = 512MB # amount of data between renegotiations
#password_encryption = on
#db_user_namespace = off

# Kerberos and GSSAPI
#krb_server_keyfile = ''
#krb_srvname = 'postgres' # (Kerberos only)
#krb_caseins_users = off

# - TCP Keepalives -
# see "man 7 tcp" for details

#tcp_keepalives_idle = 0 # TCP_KEEPIDL, in seconds;
# 0 selects the system default
#tcp_keepalives_interval = 0 # TCP_KEEPINTVL, in seconds;
# 0 selects the system default
#tcp_keepalives_count = 0 # TCP_KEEPCNT;
# 0 selects the system default

#-----
# RESOURCE USAGE (except WAL)
#-----

# - Memory -

# NOTE: This has been modified by EnterpriseDB's Tuning Wizard on 2012/11/10 16:24:05
# Original Value for "shared_buffers" was "32MB"
shared_buffers = 1024 # min 128kB
# (change requires restart)
#temp_buffers = 8MB # min 800kB
#max_prepared_transactions = 0 # zero disables the feature
# (change requires restart)
# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 1MB # min 64kB

```

```
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
work_mem = 1024                # min 64kB
#maintenance_work_mem = 16MB  # min 1MB
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
maintenance_work_mem = 16384  # min 1MB
#max_stack_depth = 2MB        # min 100kB

# - Kernel Resource Usage -

#max_files_per_process = 1000 # min 25
#                          # (change requires restart)
#shared_preload_libraries = '' # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0ms      # 0-100 milliseconds
#vacuum_cost_page_hit = 1     # 0-10000 credits
#vacuum_cost_page_miss = 10   # 0-10000 credits
#vacuum_cost_page_dirty = 20  # 0-10000 credits
#vacuum_cost_limit = 200      # 1-10000 credits

# - Background Writer -

#bgwriter_delay = 200ms       # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100  # 0-1000 max buffers written/round
#bgwriter_lru_multiplier = 2.0 # 0-10.0 multiplier on buffers scanned/round

# - Asynchronous Behavior -

#effective_io_concurrency = 1  # 1-1000. 0 disables prefetching
#-----
# WRITE AHEAD LOG
#-----
# - Settings -

#fsync = on                    # turns forced synchronization on or off
#synchronous_commit = on      # immediate fsync at commit
#wal_sync_method = fsync       # the default is the first option
                                # supported by the operating system:
                                #   open_datasync
                                #   fdatasync (default on Linux)
                                #   fsync
                                #   fsync_writethrough
                                #   open_sync
#full_page_writes = on        # recover from partial page writes
#wal_buffers = 64kB           # min 32kB
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
#wal_buffers = 256            # min 32kB
                                # (change requires restart)
#wal_writer_delay = 200ms     # 1-10000 milliseconds

#commit_delay = 0              # range 0-100000, in microseconds
#commit_siblings = 5          # range 1-1000

# - Checkpoints -

#checkpoint_segments = 3      # in logfile segments, min 1, 16MB each
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
```

```
checkpoint_segments = 8          # in logfile segments, min 1, 16MB each
#checkpoint_timeout = 5min        # range 30s-1h
#checkpoint_completion_target = 0.5 # checkpoint target duration, 0.0 - 1.0
#checkpoint_warning = 30s        # 0 disables

# - Archiving -

#archive_mode = off              # allows archiving to be done
                                # (change requires restart)
#archive_command = ''           # command to use to archive a logfile segment
#archive_timeout = 0            # force a logfile segment switch after this
                                # number of seconds; 0 disables

#-----
# QUERY TUNING
#-----

# - Planner Method Configuration -

#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_indexscan = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on

# - Planner Cost Constants -

#seq_page_cost = 1.0             # measured on an arbitrary scale
#random_page_cost = 4.0         # same scale as above
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard
random_page_cost = 4.0         # same scale as above
#cpu_tuple_cost = 0.01          # same scale as above
#cpu_index_tuple_cost = 0.005   # same scale as above
#cpu_operator_cost = 0.0025     # same scale as above
#effective_cache_size = 128MB
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard
effective_cache_size = 55328

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
#geqo_effort = 5                 # range 1-10
#geqo_pool_size = 0             # selects default based on effort
#geqo_generations = 0           # selects default based on effort
#geqo_selection_bias = 2.0     # range 1.5-2.0

# - Other Planner Options -

#default_statistics_target = 100 # range 1-10000
#constraint_exclusion = partition # on, off, or partition
#cursor_tuple_fraction = 0.1    # range 0.0-1.0
#from_collapse_limit = 8
#join_collapse_limit = 8       # 1 disables collapsing of explicit
                                # JOIN clauses
```

```
-----
# ERROR REPORTING AND LOGGING
#-----
# - Where to Log -

log_destination = 'stderr'          # Valid values are combinations of
                                     # stderr, csvlog, syslog and eventlog,
                                     # depending on platform.  csvlog
                                     # requires logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on              # Enable capturing of stderr and csvlog
                                     # into log files. Required to be on for
                                     # csvlogs.
                                     # (change requires restart)

# These are only used if logging_collector is on:
log_directory = 'pg_log'           # directory where log files are written,
                                     # can be absolute or relative to PGDATA
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log' # log file name pattern,
                                     # can include strftime() escapes
log_truncate_on_rotation = off     # If on, an existing log file of the
                                     # same name as the new log file will be
                                     # truncated rather than appended to.
                                     # But such truncation only occurs on
                                     # time-driven rotation, not on restarts
                                     # or size-driven rotation.  Default is
                                     # off, meaning append to existing files
                                     # in all cases.
log_rotation_age = 1d              # Automatic rotation of logfiles will
                                     # happen after that time.  0 disables.
log_rotation_size = 10MB          # Automatic rotation of logfiles will
                                     # happen after that much log output.
                                     # 0 disables.

# These are relevant when logging to syslog:
syslog_facility = 'LOCAL0'
syslog_ident = 'postgres'

silent_mode = off                  # Run server silently.
                                     # DO NOT USE without syslog or
                                     # logging_collector
                                     # (change requires restart)

# - When to Log -

client_min_messages = notice      # values in order of decreasing detail:
                                     #   debug5
                                     #   debug4
                                     #   debug3
                                     #   debug2
                                     #   debug1
                                     #   log
                                     #   notice
                                     #   warning
                                     #   error

log_min_messages = warning        # values in order of decreasing detail:
                                     #   debug5
```

```
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic

#log_error_verbosity = default          # terse, default, or verbose messages

#log_min_error_statement = error # values in order of decreasing detail:
# debug5
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic (effectively off)

#log_min_duration_statement = -1 # -1 is disabled, 0 logs all statements
# and their durations, > 0 logs only
# statements running at least this number
# of milliseconds

# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_checkpoints = off
#log_connections = off
#log_disconnections = off
#log_duration = off
#log_hostname = off
log_line_prefix = '%t '                # special values:
# %u = user name
# %d = database name
# %r = remote host and port
# %h = remote host
# %p = process ID
# %t = timestamp without milliseconds
# %m = timestamp with milliseconds
# %i = command tag
# %c = session ID
# %l = session line number
# %s = session start timestamp
# %v = virtual transaction ID
# %x = transaction ID (0 if none)
# %q = stop here in non-session
```

```

#          processes
#  %% = '%'
# e.g. '<u%%d> '
#log_lock_waits = off          # log lock waits >= deadlock_timeout
#log_statement = 'none'       # none, ddl, mod, all
#log_temp_files = -1         # log temporary files equal or larger
                              # than the specified size in kilobytes;
                              # -1 disables, 0 logs all temp files
#log_timezone = unknown      # actually, defaults to TZ environment
                              # setting

#-----
# RUNTIME STATISTICS
#-----

# - Query/Index Statistics Collector -

track_activities = on
#track_counts = on
#track_functions = none      # none, pl, all
#track_activity_query_size = 1024
#update_process_title = on
#stats_temp_directory = 'pg_stat_tmp'

# - Statistics Monitoring -

#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off

#-----
# AUTOVACUUM PARAMETERS
#-----

#autovacuum = on             # Enable autovacuum subprocess? 'on'
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
autovacuum = true           # Enable autovacuum subprocess? 'on'
                              # requires track_counts to also be on.
#log_autovacuum_min_duration = -1 # -1 disables, 0 logs all actions and
                              # their durations, > 0 logs only
                              # actions running at least this number
                              # of milliseconds.
#autovacuum_max_workers = 3  # max number of autovacuum subprocesses
#autovacuum_naptime = 1min   # time between autovacuum runs
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
autovacuum_naptime = 60     # time between autovacuum runs
#autovacuum_vacuum_threshold = 50 # min number of row updates before
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
autovacuum_vacuum_threshold = 1000 # min number of row updates before
                              # vacuum
#autovacuum_analyze_threshold = 50 # min number of row updates before
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
autovacuum_analyze_threshold = 250 # min number of row updates before
                              # analyze
#autovacuum_vacuum_scale_factor = 0.2 # fraction of table size before vacuum
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard

```



```

autovacuum_vacuum_scale_factor = 0.2      # fraction of table size before vacuum
#autovacuum_analyze_scale_factor = 0.1    # fraction of table size before analyze
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard
autovacuum_analyze_scale_factor = 0.1    # fraction of table size before analyze
#autovacuum_freeze_max_age = 200000000   # maximum XID age before forced vacuum
                                           # (change requires restart)
#autovacuum_vacuum_cost_delay = 20ms      # default vacuum cost delay for
                                           # autovacuum, in milliseconds;
                                           # -1 means use vacuum_cost_delay
#autovacuum_vacuum_cost_limit = -1        # default vacuum cost limit for
                                           # autovacuum, -1 means use
                                           # vacuum_cost_limit

-----
# CLIENT CONNECTION DEFAULTS
-----

# - Statement Behavior -

#search_path = '$user',public'           # schema names
#default_tablespace = ''                 # a tablespace name, '' uses the default
#temp_tablespaces = ''                   # a list of tablespace names, '' uses
                                           # only default tablespace

#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#session_replication_role = 'origin'
#statement_timeout = 0                   # in milliseconds, 0 is disabled
#vacuum_freeze_min_age = 50000000
#vacuum_freeze_table_age = 150000000
#xmlbinary = 'base64'
#xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, dmy'
#intervalstyle = 'postgres'
#timezone = unknown                       # actually, defaults to TZ environment
                                           # setting
#timezone_abbreviations = 'Default'       # Select the set of available time zone
                                           # abbreviations.  Currently, there are
                                           #   Default
                                           #   Australia
                                           #   India
                                           # You can create your own file in
                                           # share/timezonesets/.

#extra_float_digits = 0                   # min -15, max 2
#client_encoding = sql_ascii              # actually, defaults to database
                                           # encoding

# These settings are initialized by initdb, but they can be changed.
lc_messages = 'Portuguese, Portugal'     # locale for system error message
                                           # strings
lc_monetary = 'Portuguese, Portugal'     # locale for monetary formatting
lc_numeric = 'Portuguese, Portugal'      # locale for number formatting
lc_time = 'Portuguese, Portugal'         # locale for time formatting

```

```
# default configuration for text search
default_text_search_config = 'pg_catalog.simple'

# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''

#-----
# LOCK MANAGEMENT
#-----

#deadlock_timeout = 1s
#max_locks_per_transaction = 64          # min 10
                                         # (change requires restart)

# Note: Each lock table slot uses ~270 bytes of shared memory, and there are
# max_locks_per_transaction * (max_connections + max_prepared_transactions)
# lock table slots.

#-----
# VERSION/PLATFORM COMPATIBILITY
#-----

# - Previous PostgreSQL Versions -

#add_missing_from = off
#array_nulls = on
#backslash_quote = safe_encoding # on, off, or safe_encoding
#default_with_oids = off
#escape_string_warning = on
#regex_flavor = advanced          # advanced, extended, or basic
#sql_inheritance = on
#standard_conforming_strings = off
#synchronize_seqscans = on

# - Other Platforms and Clients -

#transform_null_equals = off

#-----
# CUSTOMIZED OPTIONS
#-----

#custom_variable_classes = ''          # list of custom variable class names
```

## A.4 - Ficheiro de configuração Mixed do ficheiro postgresql.conf

```
# -----
# PostgreSQL configuration file
# -----
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes          Time units:  ms = milliseconds
#                 MB = megabytes          s           = seconds
#                 GB = gigabytes          min          = minutes
#                                           h           = hours
#                                           d           = days
#-----
# FILE LOCATIONS
#-----

# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

#data_directory = 'ConfigDir'           # use data in another directory
#                                           # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'     # host-based authentication file
#                                           # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf'  # ident configuration file
#                                           # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = '(none)'           # write an extra PID file
#                                           # (change requires restart)
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -

listen_addresses = '*'                  # what IP address(es) to listen on;
#                                           # comma-separated list of addresses;
```

```

# defaults to 'localhost', '*' = all
# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = '' # (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
#bonjour_name = '' # defaults to the computer name
# (change requires restart)

# - Security and Authentication -

#authentication_timeout = 1min # 1s-600s
#ssl = off # (change requires restart)
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers
# (change requires restart)
#ssl_renegotiation_limit = 512MB # amount of data between renegotiations
#password_encryption = on
#db_user_namespace = off

# Kerberos and GSSAPI
#krb_server_keyfile = ''
#krb_srvname = 'postgres' # (Kerberos only)
#krb_caseins_users = off

# - TCP Keepalives -
# see "man 7 tcp" for details

#tcp_keepalives_idle = 0 # TCP_KEEPIDLE, in seconds;
# 0 selects the system default
#tcp_keepalives_interval = 0 # TCP_KEEPINTVL, in seconds;
# 0 selects the system default
#tcp_keepalives_count = 0 # TCP_KEEPCNT;
# 0 selects the system default

#-----
# RESOURCE USAGE (except WAL)
#-----

# - Memory -

# NOTE: This has been modified by EnterpriseDB's Tuning Wizard on 2012/11/10 16:31:27
# Original Value for "shared_buffers" was "32MB"
shared_buffers = 69642 # min 128kB
# (change requires restart)
#temp_buffers = 8MB # min 800kB
#max_prepared_transactions = 0 # zero disables the feature
# (change requires restart)
# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 1MB # min 64kB
# NOTE: This parameter is been added by EnterpriseDB's Tuning
work_mem = 11039 # min 64kB
#maintenance_work_mem = 16MB # min 1MB

```

```
# NOTE: This parameter is been added by EnterpriseDB's Tuning
maintenance_work_mem = 109516          # min 1MB
#max_stack_depth = 2MB                 # min 100kB

# - Kernel Resource Usage -

#max_files_per_process = 1000          # min 25
#                                     # (change requires restart)
#shared_preload_libraries = ''         # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0ms               # 0-100 milliseconds
#vacuum_cost_page_hit = 1              # 0-10000 credits
#vacuum_cost_page_miss = 10           # 0-10000 credits
#vacuum_cost_page_dirty = 20          # 0-10000 credits
#vacuum_cost_limit = 200               # 1-10000 credits

# - Background Writer -

#bgwriter_delay = 200ms                # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100           # 0-1000 max buffers written/round
#bgwriter_lru_multiplier = 2.0         # 0-10.0 multiplier on buffers scanned/round

# - Asynchronous Behavior -

#effective_io_concurrency = 1          # 1-1000. 0 disables prefetching

#-----
# WRITE AHEAD LOG
#-----

# - Settings -

#fsync = on                            # turns forced synchronization on or off
#synchronous_commit = on                # immediate fsync at commit
#wal_sync_method = fsync                # the default is the first option
#                                     # supported by the operating system:
#                                     #   open_datasync
#                                     #   fdatasync (default on Linux)
#                                     #   fsync
#                                     #   fsync_writethrough
#                                     #   open_sync
#full_page_writes = on                 # recover from partial page writes
#wal_buffers = 64kB                    # min 32kB
# NOTE: This parameter is been added by EnterpriseDB's Tuning
#wal_buffers = 256                      # min 32kB
#                                     # (change requires restart)
#wal_writer_delay = 200ms               # 1-10000 milliseconds

#commit_delay = 0                      # range 0-100000, in microseconds
#commit_siblings = 5                   # range 1-1000

# - Checkpoints -

#checkpoint_segments = 3                # in logfile segments, min 1, 16MB each
# NOTE: This parameter is been added by EnterpriseDB's Tuning
```

```
checkpoint_segments = 64          # in logfile segments, min 1, 16MB each
#checkpoint_timeout = 5min        # range 30s-1h
#checkpoint_completion_target = 0.5 # checkpoint target duration, 0.0 - 1.0
#checkpoint_warning = 30s        # 0 disables

# - Archiving -

#archive_mode = off              # allows archiving to be done
                                # (change requires restart)
#archive_command = ''           # command to use to archive a logfile segment
#archive_timeout = 0            # force a logfile segment switch after this
                                # number of seconds; 0 disables

#-----
# QUERY TUNING
#-----

# - Planner Method Configuration -

#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_indexscan = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on

# - Planner Cost Constants -

#seq_page_cost = 1.0             # measured on an arbitrary scale
#random_page_cost = 4.0         # same scale as above
# NOTE: This parameter is been added by EnterpriseDB's Tuning
random_page_cost = 4.0         # same scale as above
#cpu_tuple_cost = 0.01         # same scale as above
#cpu_index_tuple_cost = 0.005  # same scale as above
#cpu_operator_cost = 0.0025    # same scale as above
#effective_cache_size = 128MB
# NOTE: This parameter is been added by EnterpriseDB's Tuning
effective_cache_size = 43062

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
#geqo_effort = 5                # range 1-10
#geqo_pool_size = 0             # selects default based on effort
#geqo_generations = 0          # selects default based on effort
#geqo_selection_bias = 2.0     # range 1.5-2.0

# - Other Planner Options -

#default_statistics_target = 100 # range 1-10000
#constraint_exclusion = partition # on, off, or partition
#cursor_tuple_fraction = 0.1    # range 0.0-1.0
#from_collapse_limit = 8
```

```
#join_collapse_limit = 8          # 1 disables collapsing of explicit
                                   # JOIN clauses

#-----
# ERROR REPORTING AND LOGGING
#-----

# - Where to Log -

log_destination = 'stderr'         # Valid values are combinations of
                                   # stderr, csvlog, syslog and eventlog,
                                   # depending on platform.  csvlog
                                   # requires logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on             # Enable capturing of stderr and csvlog
                                   # into log files. Required to be on for
                                   # csvlogs.
                                   # (change requires restart)

# These are only used if logging_collector is on:
log_directory = 'pg_log'          # directory where log files are written,
                                   # can be absolute or relative to PGDATA
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log' # log file name pattern,
                                   # can include strftime() escapes
log_truncate_on_rotation = off    # If on, an existing log file of the
                                   # same name as the new log file will be
                                   # truncated rather than appended to.
                                   # But such truncation only occurs on
                                   # time-driven rotation, not on restarts
                                   # or size-driven rotation.  Default is
                                   # off, meaning append to existing files
                                   # in all cases.
log_rotation_age = 1d             # Automatic rotation of logfiles will
                                   # happen after that time.  0 disables.
log_rotation_size = 10MB         # Automatic rotation of logfiles will
                                   # happen after that much log output.
                                   # 0 disables.

# These are relevant when logging to syslog:
syslog_facility = 'LOCAL0'
syslog_ident = 'postgres'

silent_mode = off                 # Run server silently.
                                   # DO NOT USE without syslog or
                                   # logging_collector
                                   # (change requires restart)

# - When to Log -

client_min_messages = notice     # values in order of decreasing detail:
                                   #   debug5
                                   #   debug4
                                   #   debug3
                                   #   debug2
                                   #   debug1
```

```
# log
# notice
# warning
# error

#log_min_messages = warning # values in order of decreasing detail:
# debug5
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic

#log_error_verbosity = default # terse, default, or verbose messages

#log_min_error_statement = error # values in order of decreasing detail:
# debug5
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic (effectively off)

#log_min_duration_statement = -1 # -1 is disabled, 0 logs all statements
# and their durations, > 0 logs only
# statements running at least this number
# of milliseconds

# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_checkpoints = off
#log_connections = off
#log_disconnections = off
#log_duration = off
#log_hostname = off
log_line_prefix = '%t ' # special values:
# %u = user name
# %d = database name
# %r = remote host and port
# %h = remote host
# %p = process ID
```



```

# %t = timestamp without milliseconds
# %m = timestamp with milliseconds
# %i = command tag
# %c = session ID
# %l = session line number
# %s = session start timestamp
# %v = virtual transaction ID
# %x = transaction ID (0 if none)
# %q = stop here in non-session
#       processes
# %% = '%'
# e.g. '<u%%d> '

#log_lock_waits = off           # log lock waits >= deadlock_timeout
#log_statement = 'none'        # none, ddl, mod, all
#log_temp_files = -1          # log temporary files equal or larger
                               # than the specified size in kilobytes;
                               # -1 disables, 0 logs all temp files
#log_timezone = unknown       # actually, defaults to TZ environment
                               # setting

#-----
# RUNTIME STATISTICS
#-----

# - Query/Index Statistics Collector -

track_activities = on
#track_counts = on
#track_functions = none        # none, pl, all
#track_activity_query_size = 1024
#update_process_title = on
#stats_temp_directory = 'pg_stat_tmp'

# - Statistics Monitoring -

#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off

#-----
# AUTOVACUUM PARAMETERS
#-----

#autovacuum = on              # Enable autovacuum subprocess? 'on'
# NOTE: This parameter is been added by EnterpriseDB's Tuning
autovacuum = true             # Enable autovacuum subprocess? 'on'
                               # requires track_counts to also be on.
#log_autovacuum_min_duration = -1 # -1 disables, 0 logs all actions and
                               # their durations, > 0 logs only
                               # actions running at least this number
                               # of milliseconds.
#autovacuum_max_workers = 3   # max number of autovacuum subprocesses
#autovacuum_naptime = 1min    # time between autovacuum runs
# NOTE: This parameter is been added by EnterpriseDB's Tuning

```

```

autovacuum_naptime = 60          # time between autovacuum runs
#autovacuum_vacuum_threshold = 50 # min number of row updates before
# NOTE: This parameter is been added by EnterpriseDB's Tuning
autovacuum_vacuum_threshold = 1000      # min number of row updates before
                                         # vacuum
#autovacuum_analyze_threshold = 50      # min number of row updates before
# NOTE: This parameter is been added by EnterpriseDB's Tuning
autovacuum_analyze_threshold = 250      # min number of row updates before
                                         # analyze
#autovacuum_vacuum_scale_factor = 0.2    # fraction of table size before vacuum
# NOTE: This parameter is been added by EnterpriseDB's Tuning
autovacuum_vacuum_scale_factor = 0.2    # fraction of table size before vacuum
#autovacuum_analyze_scale_factor = 0.1   # fraction of table size before analyze
# NOTE: This parameter is been added by EnterpriseDB's Tuning
autovacuum_analyze_scale_factor = 0.1   # fraction of table size before analyze
#autovacuum_freeze_max_age = 200000000  # maximum XID age before forced vacuum
                                         # (change requires restart)
#autovacuum_vacuum_cost_delay = 20ms     # default vacuum cost delay for
                                         # autovacuum, in milliseconds;
                                         # -1 means use vacuum_cost_delay
#autovacuum_vacuum_cost_limit = -1       # default vacuum cost limit for
                                         # autovacuum, -1 means use
                                         # vacuum_cost_limit

#-----
# CLIENT CONNECTION DEFAULTS
#-----

# - Statement Behavior -

#search_path = '$user',public'          # schema names
#default_tablespace = ''                # a tablespace name, '' uses the default
#temp_tablespaces = ''                  # a list of tablespace names, '' uses
                                         # only default tablespace

#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#session_replication_role = 'origin'
#statement_timeout = 0                  # in milliseconds, 0 is disabled
#vacuum_freeze_min_age = 50000000
#vacuum_freeze_table_age = 150000000
#xmlbinary = 'base64'
#xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, dmy'
#intervalstyle = 'postgres'
#timezone = unknown                     # actually, defaults to TZ environment
                                         # setting
#timezone_abbreviations = 'Default'     # Select the set of available time zone
                                         # abbreviations.  Currently, there are
                                         #   Default
                                         #   Australia
                                         #   India
                                         # You can create your own file in
                                         # share/timezonesets/.

```

```
#extra_float_digits = 0          # min -15, max 2
#client_encoding = sql_ascii     # actually, defaults to database
                                  # encoding

# These settings are initialized by initdb, but they can be changed.
lc_messages = 'Portuguese, Portugal' # locale for system error message
                                  # strings
lc_monetary = 'Portuguese, Portugal' # locale for monetary formatting
lc_numeric = 'Portuguese, Portugal'  # locale for number formatting
lc_time = 'Portuguese, Portugal'     # locale for time formatting

# default configuration for text search
default_text_search_config = 'pg_catalog.simple'

# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''

#-----
# LOCK MANAGEMENT
#-----

#deadlock_timeout = 1s
#max_locks_per_transaction = 64      # min 10
                                       # (change requires restart)
# Note: Each lock table slot uses ~270 bytes of shared memory, and there are
# max_locks_per_transaction * (max_connections + max_prepared_transactions)
# lock table slots.

#-----
# VERSION/PLATFORM COMPATIBILITY
#-----

# - Previous PostgreSQL Versions -

#add_missing_from = off
#array_nulls = on
#backslash_quote = safe_encoding # on, off, or safe_encoding
#default_with_oids = off
#escape_string_warning = on
#regex_flavor = advanced          # advanced, extended, or basic
#sql_inheritance = on
#standard_conforming_strings = off
#synchronize_seqscans = on

# - Other Platforms and Clients -

#transform_null_equals = off

#-----
# CUSTOMIZED OPTIONS
#-----

#custom_variable_classes = ''      # list of custom variable class names
```

## A.5 - Ficheiro de configuração Dedicated do ficheiro postgresql.conf

```
# -----
# PostgreSQL configuration file
# -----
#
# This file consists of lines of the form:
#
#   name = value
#
# (The "=" is optional.)  Whitespace may be used.  Comments are introduced with
# "#" anywhere on a line.  The complete list of parameter names and allowed
# values can be found in the PostgreSQL documentation.
#
# The commented-out settings shown in this file represent the default values.
# Re-commenting a setting is NOT sufficient to revert it to the default value;
# you need to reload the server.
#
# This file is read on server startup and when the server receives a SIGHUP
# signal.  If you edit the file on a running system, you have to SIGHUP the
# server for the changes to take effect, or use "pg_ctl reload".  Some
# parameters, which are marked below, require a server shutdown and restart to
# take effect.
#
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on".  Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes          Time units:  ms = milliseconds
#                 MB = megabytes          s           = seconds
#                 GB = gigabytes          min          = minutes
#                                           h           = hours
#                                           d           = days
#-----
# FILE LOCATIONS
#-----
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.

#data_directory = 'ConfigDir'           # use data in another directory
#                                           # (change requires restart)
#hba_file = 'ConfigDir/pg_hba.conf'     # host-based authentication file
#                                           # (change requires restart)
#ident_file = 'ConfigDir/pg_ident.conf'  # ident configuration file
#                                           # (change requires restart)
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = '(none)'           # write an extra PID file
#                                           # (change requires restart)
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -

listen_addresses = '*'                  # what IP address(es) to listen on;
#                                           # comma-separated list of addresses;
#                                           # defaults to 'localhost', '*' = all
```

```

# (change requires restart)
port = 5432 # (change requires restart)
max_connections = 100 # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3 # (change requires restart)
#unix_socket_directory = '' # (change requires restart)
#unix_socket_group = '' # (change requires restart)
#unix_socket_permissions = 0777 # begin with 0 to use octal notation
# (change requires restart)
#bonjour_name = '' # defaults to the computer name
# (change requires restart)

# - Security and Authentication -

#authentication_timeout = 1min # 1s-600s
#ssl = off # (change requires restart)
#ssl_ciphers = 'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH' # allowed SSL ciphers
# (change requires restart)
#ssl_renegotiation_limit = 512MB # amount of data between renegotiations
#password_encryption = on
#db_user_namespace = off

# Kerberos and GSSAPI
#krb_server_keyfile = ''
#krb_srvname = 'postgres' # (Kerberos only)
#krb_caseins_users = off

# - TCP Keepalives -
# see "man 7 tcp" for details

#tcp_keepalives_idle = 0 # TCP_KEEPIDLE, in seconds;
# 0 selects the system default
#tcp_keepalives_interval = 0 # TCP_KEEPINTVL, in seconds;
# 0 selects the system default
#tcp_keepalives_count = 0 # TCP_KEEPCNT;
# 0 selects the system default

#-----
# RESOURCE USAGE (except WAL)
#-----

# - Memory -

# NOTE: This has been modified by EnterpriseDB's Tuning Wizard
# Original Value for "shared_buffers" was "32MB"
shared_buffers = 69642 # min 128kB
# (change requires restart)
#temp_buffers = 8MB # min 800kB
#max_prepared_transactions = 0 # zero disables the feature
# (change requires restart)
# Note: Increasing max_prepared_transactions costs ~600 bytes of shared memory
# per transaction slot, plus lock space (see max_locks_per_transaction).
# It is not advisable to set max_prepared_transactions nonzero unless you
# actively intend to use prepared transactions.
#work_mem = 1MB # min 64kB
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard

```

```

work_mem = 11039          # min 64kB
#maintenance_work_mem = 16MB      # min 1MB
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
maintenance_work_mem = 109516     # min 1MB
#max_stack_depth = 2MB           # min 100kB

# - Kernel Resource Usage -

#max_files_per_process = 1000     # min 25
#                               # (change requires restart)
#shared_preload_libraries = ''    # (change requires restart)

# - Cost-Based Vacuum Delay -

#vacuum_cost_delay = 0ms         # 0-100 milliseconds
#vacuum_cost_page_hit = 1       # 0-10000 credits
#vacuum_cost_page_miss = 10     # 0-10000 credits
#vacuum_cost_page_dirty = 20    # 0-10000 credits
#vacuum_cost_limit = 200        # 1-10000 credits

# - Background Writer -

#bgwriter_delay = 200ms         # 10-10000ms between rounds
#bgwriter_lru_maxpages = 100    # 0-1000 max buffers written/round
#bgwriter_lru_multiplier = 2.0  # 0-10.0 multiplier on buffers scanned/round

# - Asynchronous Behavior -

#effective_io_concurrency = 1    # 1-1000. 0 disables prefetching
#-----
# WRITE AHEAD LOG
#-----
# - Settings -

#fsync = on                    # turns forced synchronization on or off
#synchronous_commit = on       # immediate fsync at commit
#wal_sync_method = fsync        # the default is the first option
                                # supported by the operating system:
                                #   open_datasync
                                #   fdatasync (default on Linux)
                                #   fsync
                                #   fsync_writethrough
                                #   open_sync

#full_page_writes = on         # recover from partial page writes
#wal_buffers = 64kB           # min 32kB
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
#wal_buffers = 256            # min 32kB
                                # (change requires restart)
#wal_writer_delay = 200ms      # 1-10000 milliseconds

#commit_delay = 0              # range 0-100000, in microseconds
#commit_siblings = 5           # range 1-1000

# - Checkpoints -

#checkpoint_segments = 3       # in logfile segments, min 1, 16MB each
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
#checkpoint_segments = 64      # in logfile segments, min 1, 16MB each

```

```

#checkpoint_timeout = 5min          # range 30s-1h
#checkpoint_completion_target = 0.5 # checkpoint target duration, 0.0 - 1.0
#checkpoint_warning = 30s          # 0 disables

# - Archiving -

#archive_mode = off                # allows archiving to be done
                                   # (change requires restart)
#archive_command = ''              # command to use to archive a logfile segment
#archive_timeout = 0                # force a logfile segment switch after this
                                   # number of seconds; 0 disables

#-----
# QUERY TUNING
#-----

# - Planner Method Configuration -

#enable_bitmapscan = on
#enable_hashagg = on
#enable_hashjoin = on
#enable_indexscan = on
#enable_mergejoin = on
#enable_nestloop = on
#enable_seqscan = on
#enable_sort = on
#enable_tidscan = on

# - Planner Cost Constants -

#seq_page_cost = 1.0                # measured on an arbitrary scale
#random_page_cost = 4.0             # same scale as above
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard
#random_page_cost = 3.5             # same scale as above
#cpu_tuple_cost = 0.01              # same scale as above
#cpu_index_tuple_cost = 0.005       # same scale as above
#cpu_operator_cost = 0.0025         # same scale as above
#effective_cache_size = 128MB
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard
effective_cache_size = 101122

# - Genetic Query Optimizer -

#geqo = on
#geqo_threshold = 12
#geqo_effort = 5                     # range 1-10
#geqo_pool_size = 0                  # selects default based on effort
#geqo_generations = 0                # selects default based on effort
#geqo_selection_bias = 2.0           # range 1.5-2.0

# - Other Planner Options -

#default_statistics_target = 100     # range 1-10000
#constraint_exclusion = partition     # on, off, or partition
#cursor_tuple_fraction = 0.1         # range 0.0-1.0
#from_collapse_limit = 8
#join_collapse_limit = 8            # 1 disables collapsing of explicit
                                   # JOIN clauses

#-----
# ERROR REPORTING AND LOGGING

```

```
#-----  
# - Where to Log -  
  
log_destination = 'stderr'          # Valid values are combinations of  
                                     # stderr, csvlog, syslog and eventlog,  
                                     # depending on platform.  csvlog  
                                     # requires logging_collector to be on.  
  
# This is used when logging to stderr:  
logging_collector = on              # Enable capturing of stderr and csvlog  
                                     # into log files. Required to be on for  
                                     # csvlogs.  
                                     # (change requires restart)  
  
# These are only used if logging_collector is on:  
#log_directory = 'pg_log'          # directory where log files are written,  
                                     # can be absolute or relative to PGDATA  
#log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log' # log file name pattern,  
                                     # can include strftime() escapes  
#log_truncate_on_rotation = off     # If on, an existing log file of the  
                                     # same name as the new log file will be  
                                     # truncated rather than appended to.  
                                     # But such truncation only occurs on  
                                     # time-driven rotation, not on restarts  
                                     # or size-driven rotation.  Default is  
                                     # off, meaning append to existing files  
                                     # in all cases.  
#log_rotation_age = 1d             # Automatic rotation of logfiles will  
                                     # happen after that time.  0 disables.  
#log_rotation_size = 10MB         # Automatic rotation of logfiles will  
                                     # happen after that much log output.  
                                     # 0 disables.  
  
# These are relevant when logging to syslog:  
#syslog_facility = 'LOCAL0'  
#syslog_ident = 'postgres'  
  
#silent_mode = off                # Run server silently.  
                                     # DO NOT USE without syslog or  
                                     # logging_collector  
                                     # (change requires restart)  
  
# - When to Log -  
  
#client_min_messages = notice     # values in order of decreasing detail:  
                                     #   debug5  
                                     #   debug4  
                                     #   debug3  
                                     #   debug2  
                                     #   debug1  
                                     #   log  
                                     #   notice  
                                     #   warning  
                                     #   error  
  
#log_min_messages = warning       # values in order of decreasing detail:  
                                     #   debug5
```



```
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic

#log_error_verbosity = default      # terse, default, or verbose messages

#log_min_error_statement = error # values in order of decreasing detail:
# debug5
# debug4
# debug3
# debug2
# debug1
# info
# notice
# warning
# error
# log
# fatal
# panic (effectively off)

#log_min_duration_statement = -1 # -1 is disabled, 0 logs all statements
# and their durations, > 0 logs only
# statements running at least this number
# of milliseconds

# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = on
#log_checkpoints = off
#log_connections = off
#log_disconnections = off
#log_duration = off
#log_hostname = off
log_line_prefix = '%t '           # special values:
# %u = user name
# %d = database name
# %r = remote host and port
# %h = remote host
# %p = process ID
# %t = timestamp without milliseconds
# %m = timestamp with milliseconds
# %i = command tag
# %c = session ID
# %l = session line number
# %s = session start timestamp
# %v = virtual transaction ID
```

```

# %x = transaction ID (0 if none)
# %q = stop here in non-session
# processes
# %% = '%'
# e.g. '<u%%d> '
#log_lock_waits = off # log lock waits >= deadlock_timeout
#log_statement = 'none' # none, ddl, mod, all
#log_temp_files = -1 # log temporary files equal or larger
# than the specified size in kilobytes;
# -1 disables, 0 logs all temp files
#log_timezone = unknown # actually, defaults to TZ environment
# setting

#-----
# RUNTIME STATISTICS
#-----

# - Query/Index Statistics Collector -

track_activities = on
#track_counts = on
#track_functions = none # none, pl, all
#track_activity_query_size = 1024
#update_process_title = on
#stats_temp_directory = 'pg_stat_tmp'

# - Statistics Monitoring -

#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off

#-----
# AUTOVACUUM PARAMETERS
#-----

#autovacuum = on # Enable autovacuum subprocess? 'on'
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
autovacuum = true # Enable autovacuum subprocess? 'on'
# requires track_counts to also be on.
#log_autovacuum_min_duration = -1 # -1 disables, 0 logs all actions and
# their durations, > 0 logs only
# actions running at least this number
# of milliseconds.
#autovacuum_max_workers = 3 # max number of autovacuum subprocesses
#autovacuum_naptime = 1min # time between autovacuum runs
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
autovacuum_naptime = 60 # time between autovacuum runs
#autovacuum_vacuum_threshold = 50 # min number of row updates before
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard
autovacuum_vacuum_threshold = 1000 # min number of row updates before
# vacuum
#autovacuum_analyze_threshold = 50 # min number of row updates before
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wiard

```

```

autovacuum_analyze_threshold = 250      # min number of row updates before
                                         # analyze
#autovacuum_vacuum_scale_factor = 0.2    # fraction of table size before vacuum
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard
autovacuum_vacuum_scale_factor = 0.2    # fraction of table size before vacuum
#autovacuum_analyze_scale_factor = 0.1   # fraction of table size before analyze
# NOTE: This parameter is been added by EnterpriseDB's Tuning Wizard
autovacuum_analyze_scale_factor = 0.1   # fraction of table size before analyze
#autovacuum_freeze_max_age = 200000000  # maximum XID age before forced vacuum
                                         # (change requires restart)
#autovacuum_vacuum_cost_delay = 20ms     # default vacuum cost delay for
                                         # autovacuum, in milliseconds;
                                         # -1 means use vacuum_cost_delay
#autovacuum_vacuum_cost_limit = -1       # default vacuum cost limit for
                                         # autovacuum, -1 means use
                                         # vacuum_cost_limit

#-----
# CLIENT CONNECTION DEFAULTS
#-----

# - Statement Behavior -

#search_path = '$user',public'          # schema names
#default_tablespace = ''                # a tablespace name, '' uses the default
#temp_tablespaces = ''                  # a list of tablespace names, '' uses
                                         # only default tablespace

#check_function_bodies = on
#default_transaction_isolation = 'read committed'
#default_transaction_read_only = off
#session_replication_role = 'origin'
#statement_timeout = 0                  # in milliseconds, 0 is disabled
#vacuum_freeze_min_age = 50000000
#vacuum_freeze_table_age = 150000000
#xmlbinary = 'base64'
#xmloption = 'content'

# - Locale and Formatting -

datestyle = 'iso, dmy'
#intervalstyle = 'postgres'
#timezone = unknown                     # actually, defaults to TZ environment
                                         # setting
#timezone_abbreviations = 'Default'     # Select the set of available time zone
                                         # abbreviations.  Currently, there are
                                         #   Default
                                         #   Australia
                                         #   India
                                         # You can create your own file in
                                         # share/timezonesets/.

#extra_float_digits = 0                  # min -15, max 2
#client_encoding = sql_ascii             # actually, defaults to database
                                         # encoding

# These settings are initialized by initdb, but they can be changed.
lc_messages = 'Portuguese, Portugal'    # locale for system error message
                                         # strings
lc_monetary = 'Portuguese, Portugal'    # locale for monetary formatting
lc_numeric = 'Portuguese, Portugal'     # locale for number formatting

```

```
lc_time = 'Portuguese, Portugal'          # locale for time formatting

# default configuration for text search
default_text_search_config = 'pg_catalog.simple'

# - Other Defaults -

#dynamic_library_path = '$libdir'
#local_preload_libraries = ''

#-----
# LOCK MANAGEMENT
#-----

#deadlock_timeout = 1s
#max_locks_per_transaction = 64          # min 10
                                         # (change requires restart)

# Note: Each lock table slot uses ~270 bytes of shared memory, and there are
# max_locks_per_transaction * (max_connections + max_prepared_transactions)
# lock table slots.

#-----
# VERSION/PLATFORM COMPATIBILITY
#-----

# - Previous PostgreSQL Versions -

#add_missing_from = off
#array_nulls = on
#backslash_quote = safe_encoding # on, off, or safe_encoding
#default_with_oids = off
#escape_string_warning = on
#regex_flavor = advanced             # advanced, extended, or basic
#sql_inheritance = on
#standard_conforming_strings = off
#synchronize_seqscans = on

# - Other Platforms and Clients -

#transform_null_equals = off

#-----
# CUSTOMIZED OPTIONS
#-----

#custom_variable_classes = ''          # list of custom variable class names
```