



**UNIVERSIDADE DE ÉVORA**

**Escola de Ciências e Tecnologias**

Departamento de Informática

**Aplicação de Técnicas de Business  
Intelligence a Base de Dados  
Prosopográficas**

**Adão Baptista Pereira Lopes**

Orientação: Carlos Pampulim Caldeira

**Mestrado em Engenharia Informática**

Informática

Dissertação

Évora, 2017





**UNIVERSIDADE DE ÉVORA**

**Escola de Ciências e Tecnologias**

Departamento de Informática

**Aplicação de Técnicas de Business  
Intelligence a Base de Dados  
Prosopográficas**

**Adão Baptista Pereira Lopes**

Orientação: Carlos Pampulim Caldeira

**Mestrado em Engenharia Informática**

Informática

Dissertação

Évora, 2017



*«O conjunto de dados gera um registo, o conjunto de registos gera uma informação e o conjunto de informações gera o conhecimento.»*

**Autor Desconhecido**



A Presente proposta de Dissertação de Mestrado preenche o requisito parcial para a obtenção do grau de mestre em Engenharia Informática, na Universidade de Évora.

## Agradecimentos

*“Tudo posso naquele que me fortalece”*  
Filipenses 4:13

A realização deste projeto não teria a mesma amplitude caso não pudesse contar com o apoio, estímulo e colaboração das pessoas que, de muitas maneiras, me motivaram à concretização desta minha etapa académica. Portanto, e desde já, pretendo manifestar os meus sinceros agradecimentos a todos vocês.

Como em tudo na minha vida, dou graças constantemente a Deus Pai por todas as coisas, em nome de Jesus Cristo, pelo dom da vida, pela sua misericórdia, seu cuidado, proteção, por me amar sem merecer, e principalmente por me carregar em seus braços e fortalecer-me quando faltava forças para continuar frente as dificuldades.

Agradeço ao meu Pai (Tchimchim) e minha mãe (Luluxa), pela responsabilidade da minha educação, pela formação do meu carácter, pelo amor e amizade e empenho que desde a minha infância incentivaram-me, e demonstraram-me a importância dos estudos, sabendo de antemão que não tiveram as mesmas oportunidades que tive. Sei que estão orgulhosos de mim por ter concluído mais esta fase, e este trabalho é em parte para vós.

Meus agradecimentos também são dirigidas as minhas irmãs (Eva, Noemi e Raquel), meus sobrinhos (Eli e Gab) e meus cunhados (Ady e Eliseu), pelo apoio e amizade, agradeço a todos os amigos e amigas, familiares e colegas de mestrado, que vivenciaram os momentos de estudo companheirismo e auxílio prestados.

Especialmente gostaria de expressar meus agradecimentos pela Elsie Araujo, por tudo que fez por mim para que fosse possível a concretização de um dos maiores sonhos que almejei na minha vida, ter um mestrado.

Um muito obrigado ao meu orientador, Professor Doutor Carlos Pampulim Caldeira, que me orientou de um modo exemplar na elaboração deste trabalho, pela paciência que teve sempre para me ouvir, pela sua constante disponibilidade, incentivo, ajuda que me ofereceu nos diversos momentos nesta elaboração e pelos conselhos e sugestões dadas.

A Professora Fernanda Olival agradeço pela amizade, pela disposição, apoio e boa disposição que sempre manifestou. A todos os meus professores, que com apenas um sorriso contribuíram para eu continuar.

A todos aqueles que, embora não nomeados, me brindaram com seus inestimáveis apoios em distintos momentos e por suas presenças afectivas em inesquecíveis momentos, o meu reconhecido e carinhoso muito obrigado.

*“A todos os outros comuns mortais que comigo têm caminhado no trilho da vida”*  
**Bem hajam!**



# Índice

Agradecimentos .....	vii
Resumo .....	xix
Abstract.....	xxi
1. Introdução.....	3
1.1. Enquadramento e Motivação .....	3
1.2. Finalidade e Objetivos .....	4
1.3. Estruturação da Dissertação .....	6
2. Prosopografia .....	11
3. Business Intelligence.....	17
3.1. Conceito .....	18
3.1.1. Objetivos.....	19
3.2. Sistemas de Business Intelligence .....	19
3.2.1. Desvantagens .....	21
3.2.2. Vantagens .....	21
3.2.3. Sistemas transacionais (OLTP) .....	22
3.2.4. Sistemas Analíticas (OLAP).....	23
3.3. OLAP versus OLTP .....	24
3.4. Ferramentas de Business Intelligence .....	25
3.5. Processo de Extração Transformação e Carregamento (ETL).....	26
3.6. Dashboards e Scorecards .....	29
3.6.1. Conceito de Dashboards e Scorecards.....	29
3.6.2. As Finalidades de Dashboards e Scorecards .....	30
3.7. Cubos Multidimensionais (OLAP) .....	32
3.7.1. Tipos de Operações no Cubo Multidimensional. ....	35
3.8. Análise e Exploração dos Dados .....	37
4. O Modelo Dimensional .....	41
4.1. Modelo Inmon Versus Modelo Kimball .....	42
4.1.1. O Modelo Inmon .....	43
4.1.2. O Modelo Kimball.....	43
4.2. O que é Data Warehouse?.....	46
4.3. Objetivos e Vantagens da Data Warehouse .....	47
4.4. A Arquitetura do Data Warehouse.....	48
4.5. Tabela de Facto .....	49
4.5.1. Tipo de Tabelas de Factos .....	50

4.6.	Dimensões.....	51
4.6.1.	Dimensão Lixo .....	52
4.6.2.	Dimensões Role-Playing .....	53
4.7.	Data Mart Ocupações.....	54
4.7.1.	Matriz em Bus .....	55
4.7.2.	Implementação de Data Mart Ocupações .....	56
4.7.3.	Processo ETL da Data Mart Ocupações .....	58
4.7.1.	Hierarquias.....	59
4.8.	O Esquema em Estrela.....	62
4.9.	Conceito e Importância dos Metadados.....	64
5.	Casos de Estudo .....	69
5.1.	O Sistema SPARES .....	70
5.1.1.	Metodologia atual de exploração dos dados.....	73
5.1.2.	As Tabelas dimensões do Sistema SPARES .....	75
5.1.3.	A Tabela de Facto do Sistema SPARES .....	79
5.2.	A dimensão Data.....	81
5.3.	Aplicação da dimensão Data a um ambiente prosopográfico.....	83
5.4.	Metodologias aplicada na Inserção de dados na dimensão Data .....	84
6.	Análise de Dados.....	89
6.1.	Ferramenta para Análise dos dados. ....	89
6.2.	Sobre o Tableau .....	89
6.2.1.	Características de Tableau .....	90
6.2.2.	Instalação e Configuração de Tableau.....	91
6.2.3.	Conectar aos dados do sistema SPARES ao Tableau.....	91
6.2.4.	Ambiente de trabalho do Tableau.....	96
6.2.5.	Tipos de dados em Tableau .....	97
6.2.6.	Dimensões e Medidas .....	98
6.2.7.	Agregações em Tableau.....	99
6.2.8.	Navegação em Tableau.....	103
6.2.9.	Fluxo de projeto em Tableau .....	104
6.2.10.	Georreferenciação no Sistema SPARES.....	105
6.2.11.	Filtros em Tableau .....	108
6.3.	Resultados obtidos de análises de dados.....	110
6.3.1.	Análises de dados - Caso A .....	111
6.3.2.	Análise de dados - Caso B.....	112
6.3.3.	Análises de dados - Caso C .....	113

6.3.4.	Análise de dados - Caso D.....	114
6.3.5.	Análise de dados – Caso E .....	114
6.3.6.	Análise de dados – Caso F.....	115
6.3.7.	Análises de dados - Caso G .....	116
7.	Conclusão e Estudos Futuros .....	119
7.1.	Estudos Futuros.....	120
8.	Referências.....	123
	Anexos.....	127
A.	Script de criação da dimensão Data (SQL-Server).....	127
B.	Script de criação da dimensão Data (My-SQL).....	133
C.	Procedimento para atualização da tabela de Factos (Comissários) .....	142
D.	Atualizar Províncias .....	145
E.	Transformação das coordenadas geográficas .....	146
F.	Procedimentos auxiliares de data mart ocupações .....	147
G.	Processo ETL em data mart ocupações .....	149
H.	Criar data mart ocupação .....	151
I.	Função para procurar ocupação.....	152
J.	Atualizar ocupação .....	153
K.	Atualizar ocupação em comissários .....	154
L.	Efetuar ligação e atualização de ocupação na tabela de factos.....	155
M.	Acrescentar novas ocupações .....	156



## Índice de Figuras

Figura 1 - As desvantagens de BI. Fonte: (Quintanilha e Moraes, s.d).....	21
Figura 2 - As Vantagens de BI. Fonte: (Lima, s.d) .....	21
Figura 3 - Exemplo de uma transação num talão de caixa de um supermercado.....	22
Figura 4 - Arquitetura de apoio ao processo de BI. Fonte: (Murillo, 2016).....	25
Figura 5 - Modelo da área de extração, transformação e carregamento do DW .....	26
Figura 6 - Resultados obtidos em Dashboards. Fonte: (Abacus, s.d).....	30
Figura 7 – Resultados obtidos sobre Ocupações em Dashboard.....	31
Figura 8 - Aplicação do tipo Cubo. Fonte: (Oracle, s.d). .....	33
Figura 9 - Aplicação do tipo Cubo para o SPARES.....	34
Figura 10 - Magic quadrant for BI and analytics platforms. ....	36
Figura 11 - Magic quadrant for BI and analytics platforms. ....	37
Figura 12 - Elementos básicos da arquitetura da DW do modelo Kimball. ....	48
Figura 13 - Exemplo de dimensão Role Playing .....	53
Figura 14 - Matriz em Bus sobre o processo de análise de ocupações.....	55
Figura 15 - Agrupamento de ocupações (designação da época) por setores de atividade.....	57
Figura 16 -Estrutura hierárquica na dimensão Data .....	59
Figura 17 - Conceito de hierarquia na dimensão data .....	60
Figura 18 - Esquema em Estrela do SPARES .....	62
Figura 19 - Modelo dimensional do SPARES.....	71
Figura 20 - Molelo dimensional do SPARES com a nova dimensão temporal.....	72
Figura 21 - Exemplo 1 de forma atual de exploração de dados no SPARES.....	73
Figura 22 - Exemplo 2 de forma atual de exploração de dados no SPARES.....	74
Figura 23 - A utilização de surrogate key na dimensão Data .....	85
Figura 24 - Exemplo de registos para a dimensão Data .....	85
Figura 25 – (Link para download do Tableau) .....	91
Figura 26 – Passo 1 – Pagina inicial – Conectar ao servidor MySQL .....	92
Figura 27 - Passo 2 - escolher a base de dados «comissarios».....	93
Figura 28 - Lista de dimensões e tabela de factos do SPARES .....	93
Figura 29 – Passo 3 - Estabelecer a ligação entre tabela de factos e as dimensões.....	94
Figura 30 - Passo 4 - Estabelecer a ligação entre tabela de factos e as dimensões .....	95
Figura 31 - Preparação do dados em tempo real .....	95
Figura 32 - Ambiente de trabalho de Tableau .....	96
Figura 33 - Alterar tipo de dados de um atributo .....	97
Figura 34 - Agregação Mediano em Tableau .....	101
Figura 35 - Agregação Percentil em Tableau .....	102
Figura 36 - Fluxo do projeto em Tableau .....	105
Figura 37 – Análise de percentagem total do tipo ocupação Mercancia em cada.....	106
Figura 38 - Análise de percentagem total do tipo ocupação Mercancia em cada .....	107
Figura 39 - Filtros em medidas.....	108
Figura 40 - Filtros em dimensões .....	109
Figura 41 - Filtros em datas.....	110
Figura 42 - Caso A – Província da idade moderna.....	111
Figura 43 - Caso B – Bispados com maior área de Esino .....	112
Figura 44 – Caso C - Análise obtidos número total de.....	113
Figura 45 – Caso D - Reporting de número e percentagem .....	114

Figura 46 - Caso E - Reporting números de pessoas que nasceram em cada século.....	115
Figura 47 – Caso F – Províncias com os respectivos números .....	115
Figura 48 - Caso G – Províncias cartografado com a percentagem .....	116

## Índice de Quadros

Quadro 1 -Características OLAP vs Características OLTP. Adaptado de (Elias, s.d.) .....	24
Quadro 2 - Tipos de operações no Cubo. Adaptado de: (Ciferri e Ciferri, s.d).....	35
Quadro 3 - Tabela de Factos do SPARES, com as novas dimensões.....	49
Quadro 4 - Dimensão Localidade do SPARES .....	51
Quadro 5 - Sistema de Metadados .....	64
Quadro 6 - A dimensão Relação de Parentesco.....	75
Quadro 7 - A dimensão Tipo de Evento .....	75
Quadro 8 - A dimensão Tipo de Assinatura .....	75
Quadro 9 - A dimensão Investigador.....	75
Quadro 10 - A dimensão Tipo de Documento.....	75
Quadro 11 - A dimensão Ego .....	76
Quadro 12 - A dimensão Ator .....	76
Quadro 13 - A dimensão Localidade .....	77
Quadro 14- A dimensão Arquivo .....	77
Quadro 15 - A dimensão Geneologia .....	78
Quadro 16 - A dimensão Documento .....	78
Quadro 17 - A dimensão Ocupação - nova dimensão no SPARES.....	79
Quadro 18 - Tabela de Factos «comissários».....	79
Quadro 19 - A dimensão Data .....	81
Quadro 20 - Vista parcial de dados na dimensão Data.....	84
Quadro 21 - Tipos de dados em Tableau. Adaptado de: (Tableau, s.d) .....	97
Quadro 22 - Tipos de medidas e dimensões. ....	98



## Notação e Terminologia

A notação utilizada ao longo da dissertação segue o seguinte padrão:

- Texto entre aspas em itálico, para citações referentes a algum autor;
- Texto em negrito, para realçar uma palavra ou expressão.
- Texto em itálico e negrito, para palavras em língua estrangeira (e.g., Inglês);

## Listas de Acrónimos

APIs	Application Programming Interface
BI	Business Intelligence
CIDEHUS	Centro Interdisciplinar de História, Culturas e Sociedades
CSV	Comma Separated Values
DM	Data Mart
DW	Data Warehouse
ERP	Enterprise Resource Planning
ETL	Extract, Transform and Load
FRONT END	São as interfaces gráficas que permitem ao utilizador interagir com o programa
KPI	Key Performance Indicator
MIS	Management Information System
OLAP	Online Analytical Processing, em português Processo analítico em tempo real
OLTP	Online Transaction Processing, em português Processo transaccional em tempo real
OPEN SOURCE	Sistemas com código fonte aberto. Normalmente, estas ferramentas são de uma fácil usabilidade, são compartilhadas pelos seus desenvolvedores, sem restrição de acesso
ODBC	Open Database Connectivity
ODS	Operational Data Store
OWB	Oracle Warehouse Builder
PDI	Pentaho Data Integraion
ROI	Return On Investment
SPARES	Sistema Prosopográfico de Análise de Relações e Eventos Sociais
SQL	Structured Query Language
SSIS	SQL Server Integration Services



# Aplicação de Técnicas de Business Intelligence a Base de Dados Prosopográficas

## Resumo

A análise multidimensional de informação e a descoberta de padrões desconhecidos em dados são um vetor estruturante dos sistemas modernos de análise de dados.

Este trabalho pretende, em termos globais, apresentar um conjunto de técnicas e métodos de Business Intelligence que aplicadas a uma base de dados prosopográfica – o sistema SPARES – melhorem a forma atual de exploração de dados que é essencialmente baseada em filtros avulsos.

A análise temporal é de vital importância para Data Warehousing. A maior quantidade de problemas que surgem na resposta a dar, na análise de dados, aos utilizadores de Data Warehousing, nomeadamente aos gestores empresariais, está relacionada com a falta de uma dimensão “DATA”. Geralmente, a maioria de perguntas para tomada de decisões, tem como restrição o tempo (data), podendo-se então afirmar sem margem de dúvida que é um dos recursos mais valioso no modelo dimensional.

Feito este enquadramento, surge então a questão cerne do nosso objeto de estudo: será possível ter uma dimensão Temporal, de acordo com o ambiente de exploração de dados do sistema SPARES? Isto é de grande importância tendo em conta a multiplicidade de vetores que definem uma data em dados históricos entre os séculos XVI e XVIII. Paralelamente ao objetivo de otimizar e operacionalizar a análise de dados de relações e eventos com o suporte em informação prosopográfica, o trabalho incide particularmente na construção de uma aplicação do tipo CUBO, com a finalidade de melhorar a análise e exploração de dados no sistema SPARES.

**Palavras-chave:** Data Warehousing, Prosopografia, Metadados, Business Intelligence, Análise de Dados



# Application of Business Intelligence to Prosopographic Database

## Abstract

The multi-dimensional information analysis and the discovery of unknown patterns in data are a structural vector of modern data analysis systems.

This work intends, overall, to present a set of techniques and methods of business intelligence that applied to a prosopographic database - the SPARES system - can improve the current form of data exploration which is essentially based on single filters.

The majority number of problems that arise in the data analysis response to data warehousing users, namely to Business managers, is related with the lack of a "DATE" dimension. Generally, the majority of questions that decision making have, is the constraint of time (date), and it can then be argued beyond doubt that it is one of the most valuable features in the dimensional model.

Once this framework is in place, the main question of our object of study arises: will it be possible to have a Temporal dimension, according to the data exploration environment of the SPARES system? This is of great importance in view of the multiplicity of vectors that define a date in historical data between the sixteenth and eighteenth centuries. Parallel to the objective of optimizing and operationalizing the analysis of data of relations and events with the support in prosopographic information, the work focuses particularly on the construction of an application of the type CUBE, with the purpose of improving the analysis and exploitation of data in the SPARES system.

**Keywords:** Data Warehousing, Prosopography, Metadata, Business Intelligence, Data Analysis,



# Capítulo 1

## **Introdução**



# 1. Introdução

Este Capítulo tem por objetivo descrever e efetuar o enquadramento, e a motivação que estão subjacentes à realização deste projeto. De seguida, explicitam-se as finalidades da dissertação apresentada, e o contributo que pretende dar ao CIDEHUS. Conclui-se o capítulo com a descrição da estrutura da dissertação e dos restantes capítulos que a compõem.

## 1.1. Enquadramento e Motivação

Atualmente, nas empresas e organizações, estamos cientes do papel preponderante que as bases de dados assumem a nível de gestão e estratégia empresarial e organizacional, enquanto vetores essenciais para a necessidade de dotar o mundo empresarial e as organizações, independentemente da sua tipologia, de informação que habilite os distintos profissionais a tomarem decisões consentâneas com as exigências de obtenção de excelência da gestão, eficiência operacional e posicionamento estratégico.

O certo é que a factualidade objetiva dessa necessidade, aliada à enorme quantidade e proliferação de dados armazenada que as organizações detêm, não possibilitam por si só, simplicidade na análise de dados, dificultando sobremaneira a obtenção da eficácia e a eficiência exigíveis na prossecução dos seus objetivos.

No processo decisório, além da experiência dos gestores, é então crucial dispor de uma enorme quantidade de dados fidedignos, informações de qualidade e conhecimento, que os auxiliem nessa árdua tarefa de tomada de decisões. *“O acesso eficiente a estas informações pode significar a sobrevivência ou falência de uma organização”* (Dalfovo e Tamborlim, 2017, p. 19). Para a prestação de um serviço de qualidade é de vital importância, usufruir de informações de qualidade e eficiência elevadas, proporcionando novas e sempre renovadas oportunidades de negócio numa Sociedade que deixou de ser meramente industrial tendo passado a ser sobretudo uma Sociedade ávida de conhecimento.

Tendo em conta o cenário atual e globalizado do mundo empresarial, é facto assente que a competitividade hoje em dia é muito elevada, fruto de desafios constantes que acabam por ser geradores também de um investimento contínuo em tecnologia de informação no mercado onde as empresas e organizações atuam. Torna-se assim inegável que as empresas e a multiplicidade de organizações existentes, adotem como premissas de atuação a agilidade e assertividade na tomada de decisões. Consequentemente, e de certo modo, elas acabam por estar subordinadas a recolher, analisar e compreender os dados, com a finalidade de gerar informações vantajosas, servindo de um sustentáculo aos processos decisórios (Dalfovo e Tamborlim, 2017, p. 19).

Assim, contribuindo para identificar as oportunidades e posicionar as organizações estrategicamente, adquirindo maior competitividade, permitindo-lhes maximizar os lucros e minimizar os riscos de negócios nas tomadas de decisões considera-se que *“Neste contexto, os sistemas de data warehousing assumem um papel vital no planeamento das atividades empresariais, pois devido ao seu formato especial produzem aplicações otimizadas que são*

*um auxiliar precioso no momento da escolha da melhor opção de entre o leque das alternativas possíveis”* (Caldeira, 2012, p. 19).

Gomes (2011, p. 3) cita Inmon<sup>1</sup>, que definiu data warehousing como sendo uma coleção de dados relacionados a alguma área da empresa, organizados para dar suporte à decisão, e baseados nas necessidades de um determinado departamento. Segundo Kimball (citado por Gomes (2011, p. 3)) o data warehouse é definido como:

*"(...) a copy of transaction data specifically structured for query and analysis."*

Enquadrada nesta problemática, surgiu a motivação subjacente ao projeto da dissertação e que tem como finalidade específica o estudo, aplicações de técnicas de Business Intelligence a base de dados prosopográficos, para oferecer um contributo de relevo aos investigadores de Centro Interdisciplinar de História, Culturas e Sociedades, nas análises de dados sobre os comissários do Santo Ofício na Idade Moderna em Portugal.

A organização em causa, possui um repositório de dados com suporte prosopográfico, e identificou a falta de mecanismos e ferramentas potentes que possibilitem uma maior acessibilidade e exploração dos dados. É neste contexto que se propõe uma solução, que consiste basicamente em criar um aplicativo tipo cubo para análise de dados, permitindo a obtenção de informações úteis e de melhor qualidade, levando inclusivamente à descoberta de novos padrões nos dados.

## **1.2. Finalidade e Objetivos**

Este projeto tem como finalidade e objetivo primordial a investigação para aplicações de técnicas de Business Intelligence, sobre repositórios de dados com um suporte em informação prosopográfica, de modo a otimizar e operacionalizar de modo mais dinâmico a análise de dados de relações e eventos, entre os diversos atores intervenientes num determinado contexto social. Consequentemente, procura o projeto desenvolver um sistema de metadados, para integração de dados prosopográficos num silo destinado a Business Intelligence. A título complementar, destaca-se a construção de uma aplicação do tipo CUBO, para prospeção de dados no repositório desenvolvido no sistema de metadados, em conjunto com a base de dados SPARES.

O sistema SPARES é caracterizado por ser dotado de uma grande quantidade de dados prosopográficos. De modo a que informações neles contida possa ser acessível e de fácil compreensão, urge aplicar as técnicas de Business Intelligence.

Segundo Kimball e Ross (2013, p. 3) este conjunto de ferramentas proporciona que os dados sejam intuitivos e óbvios para os utilizadores, eliminando a condicionante de ter que se ter como valência pessoal muitos conhecimentos informáticos. Citando os mesmos co-autores, estes referem que as ferramentas de Sistemas de Business Intelligence para análise de

---

<sup>1</sup> Considerado como o Pai da Data Warehouse.

dados, devem ter uma enorme simplicidade, e serem de fácil usabilidade. Os resultados de uma consulta têm que ser obtidos no menor tempo possível, pelo que em suma deve ser um sistema simples e rápido.

As técnicas de Business Intelligence são de grande usabilidade, em sistemas suportados em dados quantitativos, principalmente na análise de dados para a tomada de uma decisão na organização, baseada em valores de métricas num dado intervalo de tempo. Mesmo assim, elas podem ser utilizadas em sistemas suportados com dados de natureza qualitativos, como é o caso do sistema SPARES. Aplicar as técnicas de Business Intelligence no SPARES, permitirá melhorar a consistência na apresentação de informações, e a descoberta de novos padrões de dados, algo de irrefutável valor.

O sistema SPARES, que é um repositório de dados com um suporte em informações prosopográficas, não tem ferramentas adequadas e flexíveis para as pesquisas e análise de dados, sendo a única ferramenta de pesquisa os filtros, o que dificulta de modo considerável a descoberta de novos padrões existentes nos dados.

Os filtros têm como escopo diminuir a seleção de dados apresentada na visualização de uma análise de dados. Pode servir para segmentar as explorações de dados, apresentá-los por um determinado intervalo, de acordo com o atributo de uma dimensão. O capítulo 5.1.1 remete para a explicação da metodologia atual de pesquisas no sistema SPARES, que consiste basicamente na utilização de filtros avulsos. Apresenta as suas limitações e a necessidade de se implementar uma nova ferramenta de análise de dados para esse sistema.

O objetivo deste projeto, é conceder aos utilizadores do sistema SPARES, a possibilidade de usufruírem de uma nova ferramenta de análise de dados, qualitativamente poderosa e dinâmica, permitindo manipular os dados construindo um Cubo, e tendo na sua esfera de disponibilidade o poder de alterarem as arestas e os segmentos do Cubo de acordo com as suas necessidades.

Depois da obtenção dos resultados no ecrã de análise de dados, pretende-se com os APIs <sup>2</sup> do Google Maps imprimir os resultados em mapas.

A análise temporal é de vital importância para Data Warehousing. A maior quantidade de problemas que surgem, para dar resposta na análise de dados aos utilizadores de Data Warehousing, nomeadamente aos gestores empresariais, está relacionada com a falta de uma dimensão “DATA”. Geralmente, a maioria de perguntas para tomada de decisões, tem como restrição o tempo (data), podendo-se então afirmar sem sombra de dúvida que é um dos recursos mais valioso no modelo dimensional.

Feito este enquadramento, surge então a grande questão para o nosso objeto de estudo: **será possível ter uma dimensão DATA, de acordo com o ambiente de exploração de dados do SPARES, tendo em conta a multiplicidade de vetores que definem uma data em dados históricos entre os séculos XVI e XVIII?**

---

<sup>2</sup> APIs (Application Programming Interface)

No capítulo 5.2 demonstra-se que essa dimensão constitui um contributo relevante para dar maior pujança à aplicação SPARES, permitindo ganhar outra operacionalidade, através da decomposição de uma informação em partes menores, para nos facilitar a compreensão dos dados a serem analisados. Conforme supra referenciado, hoje em dia, para uma organização ou empresa, atingir os seus objetivos e aumentar a sua eficiência operacional, é imprescindível o uso e aplicações das técnicas e métodos de Business Intelligence, o que lhe permitirá agregar valor na estratégia de negócio a implantar e uma melhoria contínua a nível da gestão organizacional e do processo decisório.

Este projeto surge alinhado com tais objetivos, através da dotação de uma ferramenta à organização identificada, que lhe permita suprir as lacunas existentes, e que lhe permita a obtenção de informações úteis com elevados padrões de qualidade.

### 1.3. Estruturação da Dissertação

Na elaboração do presente documento, que expõe todo o trabalho elaborado no âmbito do projeto de dissertação, procurou-se adaptar uma escrita simples e fundamentada. Deste modo, este documento está estruturado em sete capítulos:

**Capítulo 1:** Este capítulo tem como propósito fundamental, contextualizar e enquadrar o projeto na área em estudo, descrevendo as suas finalidades e os objetivos primordiais para a sua realização.

**Capítulo 2:** O capítulo dois evidencia a revisão sobre o estado da arte da prosopografia, a sua origem histórica, o seu conceito, os diversos projetos onde trabalham os utilizadores do sistema SPARES, fazendo referência ao projeto mais relevante, a atuação dos Comissários do Santo Ofício ou Inquisição do Santo Ofício em Portugal.

**Capítulo 3:** O capítulo três faz referência ao sistema de Business Intelligence, mencionando as suas vantagens, desvantagens, seus objetivos, a diferença entre sistemas transacionais e analíticos, assim como são contextualizadas as funções das ferramentas de extracção, transformação e carregamento de dados no sistema SPARES. As ferramentas OLAP são também caracterizadas, apresentando as diversas funções de manipulação dos cubos. expondo em resumo é apresentado o conceito de BI, caracterizando-o globalmente, e descrevendo duas das tecnologias que lhe estão associadas: DW e OLAP, as ferramentas utilizadas para o desenvolvimento deste projeto de Business Intelligence e descrevendo em particular a aplicação tipo cubo que foi utilizada neste projeto, apresentando os resultados das análises em dashboarding.

**Capítulo 4:** Neste capítulo é efetuada a abordagem do modelo dimensional, apresentando-se o novo esquema em estrela do sistema SPARES. Apresenta em relação aos sistemas de data warehousing, apresentam-se as características destes repositórios, descreve-se a modelação multidimensional de dados na concepção da arquitectura destes sistemas, Este capítulo termina com a apresentação da implementação da data mart ocupações, explicitando o seu processo de extracção, transformação e carregamento de dados e também apresenta a noção e a importância de ter os metadados num modelo dimensional.

**Capítulo 5:** Este capítulo apresenta os casos de estudos, as descrições do sistema de prosopográfico de análises de relações e eventos, apresentando o seu modelo dimensional de quando foi criado a 10 anos atrás, e o novo modelo dimensional com a inclusão da dimensão temporal. Também é parafraseado a forma atual de pesquisas e análises no sistema SPARES, de seguida é demonstrado as suas dimensões e tabela de factos, e o mais importante nesse capítulo é explicado o principal objeto de estudo que está baseado no estudo da concepção e implementação da dimensão Data no sistema SPARES.

**Capítulo 6:** O capítulo seis faz menção sobre a nova forma de análise de dados no sistema SPARES. Este capítulo funciona como um manual do aplicativo tipo cubo pra descobertas de novos padrões de dados. Apresenta a forma de instalação e configuração do Tableau de forma detalhada, explica os conceitos relacionados com o Tableau, a dimensão e as medidas, os tipos de dados, a agregação de dados, a georreferenciação em Tableau, apresenta alguns exemplos de resultados obtidos de algumas análises de dados efetua no novo sistema SPARES, demonstrando a potência de nova forma de analisar os dados no sistema prosopográfico de análise de relações e eventos.

**Capítulo 7:** Por último, o capítulo sete, apresenta as conclusões e resultados do trabalho realizado, evidenciando as dificuldades sentidas no seu desenvolvimento e fazendo-se uma referência ao trabalho futuro proposto.

Para além desses capítulos enunciados, esta dissertação faz referência a alguns anexos que complementam o projeto desenvolvido, onde consta o script das funções e dos procedimentos utilizados para o processo ETL, bem como documento sobre a instalação e a configuração de Tableau, e não menos importante, contém a bibliografia utilizada como fonte desta dissertação.



# Capítulo 2

## **Prosopografía**



## 2. Prosopografia

Não existe uma datação concreta para o aparecimento da prosopografia. Apesar da sua longa história, o primeiro uso conhecido da Prosopografia remonta à data de 1743 (Nicolet, 1970, p. 1210). Sabe-se no entanto, que já no século XVI a prosopografia tinha sido muito aplicada nos períodos da história antiga e medieval, tendo nos últimos anos esse método sido trabalhado na História Moderna e Contemporânea.

*“Esse método tem uma base simples, que consiste em definir uma população a partir de um ou alguns critérios e estabelecer assim uma descrição bibliográfica cujas nuances possibilitarão traçar um perfil da sua dinâmica social, privada, pública, cultural, ideológica ou política”* (Almeida, 2011, p. 1).

Para os historiadores da antiguidade, a prosopografia era considerada uma ciência prestadora de auxílio à história, tendo como objetivo primordial, estudar as biografias dos constituintes dos diferentes grupos das elites sociais, ou políticas de uma determinada sociedade. É de salientar o facto de existirem múltiplas definições de prosopografia, mas em todas há uma componente comum, no que diz respeito ao estudo da análise do indivíduo, em função da totalidade da qual faz parte. Etimologicamente, a prosopografia refere-se ao estudo da descrição de pessoas, mais concretamente, é o estudo da carreira de uma pessoa através da informação sobre ela, constante em fontes históricas (Almeida, 2011, p. 2) .

Como podemos constatar, a prosopografia não é algo recente, sendo possível observar, uma mudança nas expectativas e nos objetivos inerentes à prosopografia como método aplicado à pesquisa histórica (Almeida, 2011, p. 4). Ela sofreu mutações no seu sentido de uso, tendo obtido um conjunto assinalável de modificações até chegar à forma pela qual é conhecida hoje, estando inclusive sujeita a outras divergências. A título de exemplo, podemos citar que o objetivo primordial era dirigido para o estudo da descrição superficial, dos traços característicos de uma determinada personagem humana. A posteriori, a prosopografia adquiriu peculiaridade distinta ao eleger, com mais constância e regularidade, o estudo de determinados grupos sociais, nomeadamente os constituintes das realidades sociais, intelectuais, religiosas, políticas, entre outras. Não somente os seus traços e características quantitativas, como do mesmo modo os seus aspetos qualitativos devem ser considerados (Azevedo e Caminha, 2013, p. 2-3).

Segundo Bulst (2005, p. 50), o termo prosopografia significa uma *“coleção e catálogo de todas as pessoas de um grupo definido temporal e espacialmente”*. Essa definição ajuda-nos a não confundir a prosopografia com a biografia. Sendo que, na história antiga esses dois conceitos eram muito semelhantes, a diferença vital e a destacar consiste nos seus interesses diferenciáveis. De facto, conforme Bulst (2005, p. 55) argumenta, *“Enquanto a biografia visa o indivíduo; o interesse da prosopografia é o conjunto ou a totalidade, constantemente considerando o indivíduo nas suas relações com o conjunto”*. De acordo com o autor, é inócuo analisar um grupo prosopograficamente, procedendo à análise biográfica dos indivíduos, se não houver um objetivo para a posteriori se estabelecer a relação entre eles. Entretanto, na elaboração da prosopografia, há uma necessidade do uso da biografia. Considerando que a relação entre o indivíduo e o conjunto é, entre outros, um dos pontos que

dão contorno ao método e à construção de casos individuais, o objeto da biografia, é o ponto de partida para a etapa posterior de análise do grupo, onde os casos individuais são integrados pelas relações com o conjunto (Azevedo e Caminha, 2013, p. 5).

Constatamos, que na atualidade as investigações prosopográficas, estabelecem como escopo essencial a exploração de um aglomerado. Os estudos prosopográficos podem inclusive ser direcionados de modo divergentes, devendo no entanto cumprir algumas regras comuns. É o que refere o historiador Stone (1971, p. 46) na seguinte citação: “*focalizar as características comuns do passado de um grupo de atores na história através do estudo coletivo de suas vidas*”. O método empregue é o de estabelecer o universo a ser estudado e, então, formular um conjunto uniforme de questões – sobre nascimento e morte, casamento e família, origens sociais e posições económicas herdadas, lugar de residência, educação, tamanho e origem das riquezas pessoais, ocupação, religião, experiência profissional etc. Os vários tipos de informação sobre indivíduos de um dado universo são então justapostos e combinados e, em seguida, examinados por meio de variáveis significativas. Essas são testadas, tanto a partir das suas correlações internas, quanto correlacionadas com outras formas de comportamento ou ações.

Neste projeto, a prosopografia tem um papel decisivo, dado que é o suporte informacional dos dados do sistema SPARES. De destacar que se encontra relacionada com diversos projetos, alguns desses em fase de investigação, podendo a título exemplificativo serem mencionados (I) - O estudo das redes comerciais formais e informais nos mundos ibéricos em contexto global (1580-1640), (II) - A investigação sobre os pilares financeiros da Inquisição Portuguesa (1640,1773), (III) - A análise prosopográfica a respeito dos grupos intermédios em Portugal e no Império Português: as familiaturas do Santo Ofício (c. 1570-1773), (IV) – O projeto sobre os notários do secreto do Santo Ofício português (1600-1755). De registar que existem projetos de investigação que já se encontram encerrados, como por exemplo, o Inquirir da honra: comissários do Santo Ofício e das Ordens Militares em Portugal (1570-1773).

O tema fulcral para exploração e análise de dados, é a atuação dos Comissários do Santo Ofício ou Inquisição do Santo Ofício em Portugal. Na Idade Moderna, surgiu o chamado Santo Ofício, ou Inquisição do Santo Ofício, instituição judicial que tinha a responsabilidade de julgar, perseguir e punir todas as pessoas consideradas como hereges ou heterodoxos, ou seja, pessoas que não estavam de acordo com as normas ou práticas impostas pela Igreja Católica.

Em Portugal, o Santo Ofício tinha o seu poder inquisitorial estendido a todo o país e colónias, que se encontravam assim submetidos, consoante o caso, pelos seguintes tribunais: Tribunal de Coimbra (Entre Douro e Minho, Trás-os-Montes e Alto Douro e Beiras), Tribunal de Lisboa que subordinava as colónias portuguesas da Estremadura, Ribatejo, Ilhas e Ocidente e, Tribunal de Évora (Alentejo e Algarve).

De acordo com Olival (2012, p. 478), “*para ter segurança e uma certa honra e dignidade em termos gerais, era essencial ingressar no Santo Ofício ou nas Ordens dos Militares*”. Para esse efeito, era condicionante possuir-se determinadas habilitações que tipificavam a qualidade dos que queriam vincular-se ao Santo Ofício e que tinham uma rede

de Comissários. Essa rede é que constitui a fonte informacional do sistema SPARES, sendo deste modo premissa fundamental familiarizarmo-nos com o modo de atuação dos comissários, que funcionam especificamente como auxílio na compreensão de análise de dados, e descoberta de novos padrões de dados, bem como respetivos familiares do Santo Ofício, que eram pessoas que tinham vínculo com a Inquisição, colocando os seus serviços ao dispor dos inquisidores, coadjuvando-os no exercício dos seus ofícios.

“A historiografia tem admitido que a rede de comissários do Santo Ofício ter-se-ia iniciado no final da década de 1570 ou na seguinte” (Olival, 2012, p. 478). Em Julho de 1575<sup>3</sup> assistiu-se à nomeação do primeiro familiar para a Inquisição em Évora, mas a designação do primeiro comissário ocorreu apenas em 1586. Para o exercício desse cargo, não era exigível qualquer formação específica, pois consoante a prática da função, e a sua interação com o Tribunal do Santo Ofício, o comissário aumentaria paulatinamente o seu conhecimento e formação.

As tarefas inerentes ao exercício do cargo de comissário consistiam em registar denúncias e reencaminhar essas denúncias ao tribunal respetivo, prender pessoas, ouvir as testemunhas que deveriam ter um certo tipo de perfil como por exemplo serem consideradas pessoas de alta credibilidade e de confiança, não serem de raça de Mouro, nem Judeu, nem Negro, nem Mulato, nem cristão recém-convertido, nem pai, nem mãe, nem avó, ninguém que tivesse um grau de parentesco com o inquirido. Portanto, nomeadamente seriam os cristãos mais velhos, fiéis à igreja, que tinham a razão de conhecer os inquiridos, e que não eram da família do inquirido (Olival, 2012, p. 485-487).

Os comissários eram oriundos de zonas rurais do interior de Portugal. Todos esses indivíduos tinham que ser eclesiásticos, e a maioria provinha de agregados de lavradores ou de artesãos que detinham alguns meios económicos. Foram a “primeira geração que se afastara do trabalho mecânico, mediante alguns estudos, através do ingresso no clero”. (Garrido, Costa, e Duarte, 2012, p. 183-184). Os familiares dos comissários eram uma espécie de agentes inquisitórios, comumente leigos, com a função de denunciar os desviantes da fé, e se necessário efetuar a prisão quando assim fosse solicitado ou exigido por um inquisidor ou comissário. De igual modo, tinham a responsabilidade de fazer chegar o preso ao tribunal do Santo Ofício.

Os Familiares não tinham qualquer formação religiosa, mas como eram leigos, sempre que ocorria uma ausência dos comissários, e houvesse algum caso de acusação a efetuar, ou até de admitir algum desregramento, o povo recorria às famílias dos comissários. Resumindo, tinham como objetivo primordial, servir de canal de comunicação entre o tribunal de Santo Ofício e o povo.

Face ao exposto, é de suma importância compreendermos todo o funcionamento da rede de comissários, as suas respetivas ocupações, sua árvore genealógica, entre outros dados, de modo a facilitar a compreensão do sistema SPARES num todo, dado que essa rede é, conforme supra referido, o suporte informacional do SPARES.

---

<sup>3</sup> Tratou-se de Jerónimo de Torres, pedreiro, morador em Évora, abaixo do Chão das Covas, “mestre na obra de Sua Alteza”, o Inquisidor-Geral, nomeado “pello tempo que parecer soamente, e enquanto não mandarmos o contrajro” – ANTT, Inquisição de Évora, L<sup>o</sup> 146, fl. 139v (Olival, 2012, p. 479).

Podemos assim referir que, enquanto na dimensão de comissários, estão registados todos os eventos, e as relações da Inquisição do Santo Ofício, a data do acontecimento, o local, os atores envolvidos e demais informações, na dimensão de genealogia, encontramos o registo de todas as relações de parentesco existentes entre as pessoas envolvidas na rede dos comissários, assim como outros dados informacionais, citando-se a título de exemplo a referência a determinadas ocupações.

# Capítulo 3

## **Business Intelligence**



### 3. Business Intelligence

*«Obter dados certos, no momento certo, para tomar a decisão certa, para a organização»*

*Autor Desconhecido*

De acordo com Dalfovo e Tamborlim (2017, p. 28), o conceito de Business Intelligence está diretamente ligado ao ERP<sup>4</sup>, tendo tido a sua origem por volta do ano de 1970, através de MIS<sup>5</sup>.

Em meados dos anos de 1990 surge o termo Business Intelligence por Howard Dresner do Gartner Group<sup>6</sup>. Somente por volta de 2005 os sistemas de Business Intelligence passaram a incluir e ser dotados de poderosos recursos e ferramentas, de tratamentos e análise de dados. São esses recursos que aplicaremos no repositório de dados prosopográfico, de modo a simplificar o tratamento, exploração e análise de dados no sistema SPARES.

Vivemos numa sociedade globalizada, onde as mudanças ocorrem e são vivenciadas de modo frenético e muito rápido. O mercado é muito exigente, o que justifica a razão da premência no desenvolvimento de metodologias e estratégias, que possam fornecer às empresas um plano de ação e sobrevivência, que lhes permita obter competitividade estratégica, analítica e operacional. É neste contexto, que surge a aplicação das técnicas de Business Intelligence<sup>7</sup> para auxiliar os gestores nas tomadas de decisões, que se requerem exigentes, estratégicas e direcionadas para o sucesso.

O processo decisório nas organizações é, em geral, um desafio de suma importância, pelo que as tomadas das melhores decisões, exigem o menor erro possível. Para tal, é necessário que o gestor esteja apoiado em informações que representem um valor significativo, e que o conduza de modo fidedigno a uma boa tomada de decisão.

Um dos mecanismos de apoio à decisão, que tem ganho espaço no mercado e em discussões no meio científico, é o Business Intelligence (Filho, Clericuzi, Souza, e Bione, 2011, p. 2).

Atualmente as empresas e as organizações litigam cada dia mais a rapidez e diligência, os riscos e as oportunidades do mercado, bem como o aperfeiçoamento e melhoria contínua no feedback às necessidades dos clientes. Deste modo, as ferramentas de Business Intelligence exercem um papel vital, transformando enormes quantidades de dados armazenados, em sistemas de suporte para tomadas de decisões, independentemente da sua

<sup>4</sup> **ERP** (Enterprise Resource Planning), “representa os sistemas integrados de gestão empresarial cuja função é facilitar o aspecto operacional das empresas” (Primark, 2008, p. 3)

<sup>5</sup> **MIS** (Management Information System), centraliza a Gestão de Tecnologia de Informação, oferecendo eficácia e eficiência na tomada de decisão estratégica.

<sup>6</sup> **Gartner** é detentora da paternidade do termo Business Intelligence. Consultora de pesquisas de mercado na área das Tecnologias da Informação.

<sup>7</sup> **BI** é entendido como o processo de recolha, e tratamento de informações com objetivo de apoiar a gestão de um negócio.

grandeza, o que de per si constitui um inegável recurso a nível de competitividade e sustentabilidade empresarial.

### 3.1. Conceito

A nível conceptual, a definição do termo Business Intelligence ainda não obteve consenso entre os diversos autores. Embora possamos identificar termos e objetivos comuns em algumas das definições pesquisadas, verifica-se infra não ser possível determinar uma regra clara.

Howard Dresner definiu BI como sendo, *uma metodologia pela qual se estabelecem ferramentas para obter, organizar, analisar e prover acesso às informações necessárias aos tomadores de decisão das empresas para analisarem os fenómenos acerca de seus negócios* (Braghittoni, 2017, cap. 1).

Dalfovo e Tamborlim (2017, p. 29) citam Turban (2009), que define Business Inteligente:

*“com um termo «Guarda Chuva» que inclui arquiteturas, ferramentas, bancos de dados, aplicações e metodologias que em conjunto são utilizadas para extrair inteligência a partir dos dados de uma área de negócio”.*

Barbieri (2001, p. 34) descreve Business Intelligence como sendo a,

*“ (...) utilização de variadas fontes de informação para se definirem estratégias de competitividade nos negócios da empresa”.*

Forrester Research define Business Intelligence como sendo:

*“um conjunto de metodologias, processos, arquiteturas e tecnologias que transformam dados brutos em informações significativas e úteis com o objetivo de habilitar de forma estratégica, tática e operacional na tomada de decisão”.*

Termos comuns que podem ser verificados nas definições são: apoio à tomada de decisão, acesso à informação e ferramentas.

A Business Intelligence deverá assim, ser considerada no âmbito deste trabalho como um conjunto de técnicas, metodologias e conceitos, implementados através de ferramentas de software, orientadas para a análise de dados. Tal permitirá a disponibilização de informações relevantes de forma compreensível e transparente, proporcionando e apoiando os gestores nas tomadas de decisões eficazes, correndo assim o menor risco possível.

### 3.1.1. Objetivos

O objetivo primordial do Business Intelligence é, claramente poder ser um agente auxiliador dos gestores na elaboração de estratégias mais assertivas. De acordo com o analista de Business Intelligence, Souza (s.d), esta metodologia tem as seguintes finalidades:

- Aumentar o grau de assertividade, no planeamento operacional e estratégico;
- Auxiliar no procedimento de tomada de decisões;
- Minorar o dispêndio de tarefas não rentáveis;
- Ampliar e garantir a credibilidade das ações planeadas, e a qualidade dos produtos e serviços prestados;
- Fazer perdurar a competitividade perante a concorrência;
- Idealizar com visão de futuro as novas exequibilidades de negócio, de modo a ampliar a atuação no mercado;
- Perceber antecipadamente os riscos de negócios;
- Transformação de dados em informações, posteriormente em decisões, por último em ações.

### 3.2. Sistemas de Business Intelligence

Na atualidade, as organizações têm muitos desafios, entre os quais podemos destacar a avaliação da *performance* em tempo real, de maneira a reunir condições de decisão em tempo útil. Nas tomadas de decisão onde não há um sistema de apoio, o gestor decide às cegas, e essa decisão é geralmente baseada no conhecimento e na experiência do gestor, levando em consideração as decisões tomadas anteriormente. Em contraposição, dispondo o gestor de um sistema de Business Intelligence, as decisões são fundamentadas em dados históricos, geralmente armazenados no sistema transacional da empresa. Em síntese, a decisão não é tomada pela intuição do administrador onde há uma margem de risco muito elevada, mas sim tendo em conta factos concretos onde podemos constatar a diminuição significativa do risco de negócio.

Face ao exposto, o propósito dos sistemas de BI, consiste na concessão de tomada de decisões de cariz proativo, providenciando e gerando no momento adequado conhecimentos e informações necessárias ao negócio.

Segundo Santos e Ramos (2006, p. 2), os sistemas de BI *“utilizam os dados disponíveis nas organizações para disponibilizar informação relevante para a tomada de decisão. Combinam um conjunto de ferramentas de interrogação e exploração dos dados com ferramentas que permitem a geração de relatórios, para produzir informação que será posteriormente utilizada pela gestão de topo das organizações, no suporte à tomada de decisão.”* Referenciando ainda os mesmos autores, tradicionalmente este conceito estará agregado a três tecnologias ou ferramentas a considerar:

- **Data Warehouse**, enquanto repositório de dados, que armazena informações importantes para a tomada de decisão. No nosso caso de estudo é o sistema SPARES;
- **OLAP**, “*análise multidimensional, que nos permite examinar a informação sob diferentes perspectivas*” (Santos e Ramos, 2006, p. 2);
- **Data Mining**, segundo Antunes (2008, p. 63) este termo “*tem sido usado e abusado, significando a maioria das vezes todo o processo de extracção de informação habitualmente designado por KDD<sup>8</sup> e não a etapa de exploração dos dados, propriamente dita*”. Neste enquadramento, o data mining é definido como sendo o processo de obtenção e exploração de informações que se encontra obscura num repositório de dados.

Atualmente na maioria das empresas e organizações existem grandes armazéns de dados, embora a disponibilidade desses dados sofra frequentemente de diversas limitações. A fim de reverter essa situação, optam pela implementação de sistemas de BI, que estão a ser utilizados em quase todas as atividades comerciais, desde corporações de pequeno, médio e grande porte, devido à necessidade de obter respostas rápidas, otimizar o trabalho da organização, reduzir custos, eliminar tarefas duplicadas, permitir previsões de crescimento da empresa como um todo, contribuir para a elaboração de estratégias, e obter informações com qualidade, proporcionando decisões eficientes que ocorrem a cada instante (Barbieri, 2001, p. 6).

Deste modo, afirmamos que as metas fundamentais do sistema de BI, é desde logo efetuar a recolha dos dados das diversas fontes, transformar esses dados em informações, e por fim gerar conhecimento útil apoiando as tomadas de decisões.

O sistema de BI, não é nada mais, nada menos, que a conjunção da tecnologia e gestão. Deste modo, o plano estratégico consiste em transformar a informação em conhecimento útil relevante ao negócio, a fim de proporcionar assistência para tomada de decisões assertivas pelos gestores. Dessa forma, confere-se uma maior precisão e confiabilidade na decisão, e auxiliam-se as empresas a ter maior eficiência no mercado.

As ferramentas usadas na maioria dos sistemas de BI funcionam em tempo real, através do processo ETL que extrai os dados do sistema transacional, e em seguida efetua a transformação dos dados, realizando por fim o carregamento desses dados, para um sistema analítico de apoio a decisão.

Na atualidade o sistema de BI, está implementado na generalidade das empresas e organizações, desde instituições financeiras, empresas de telecomunicação, companhias aéreas, que seguem a tendência da economia globalizada, onde há uma vasta necessidade da rapidez, precisão, agilidade e abundância de informações. As técnicas e as ferramentas de BI evoluem progressivamente, não apenas ao nível das informações, mas igualmente no contexto dos fluxos de dados e do conhecimento, dado que o mercado dispõe de um elevado potencial de crescimento, e de exigências de conformidade a fatores de competitividade crescente (Barbieri, 2001, p. 7).

---

<sup>8</sup> **KDD**, do inglês Knowledge Discovery in Databases

### 3.2.1. Desvantagens

De acordo com a Figura 1, podemos observar que a maior desvantagem do sistema de Business Intelligence é o seu custo de implementação e manutenção. Existem outras desvantagens, nomeadamente o insucesso após a implementação de um sistema de BI, por falta de visão e conhecimento das ferramentas e suas potencialidades.

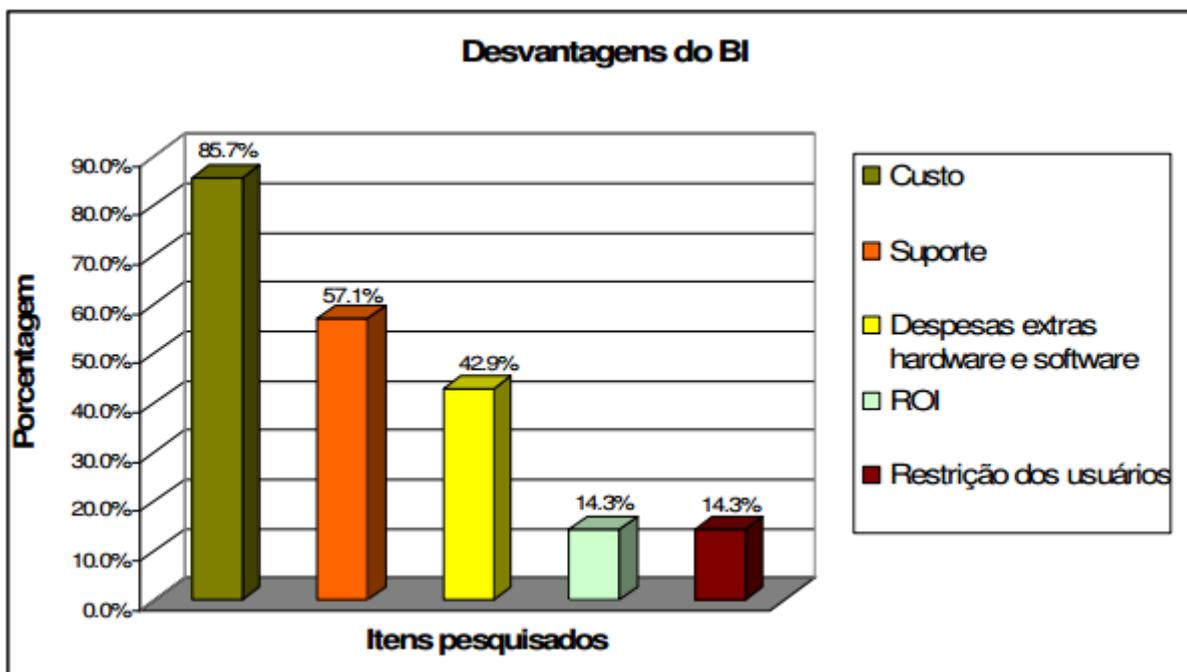


Figura 1 - As desvantagens de BI. Fonte: (Quintanilha e Moraes, s.d)

### 3.2.2. Vantagens

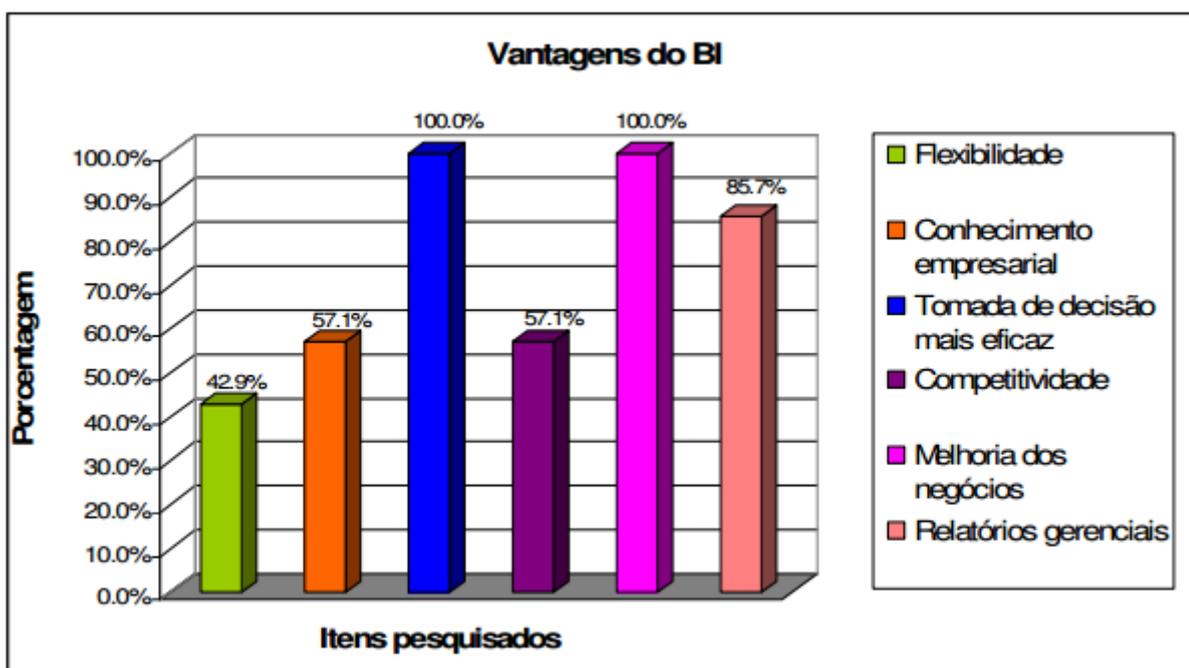


Figura 2 - As Vantagens de BI. Fonte: (Lima, s.d)

De acordo com a Figura 2, podemos observar que segundo Lima (s.d), uma pesquisa efetuada no mercado, retratando as maiores regalias e vantagens fornecidas pelas aplicações de BI no processo decisório, conclui que a tomada de decisão mais eficaz e a melhoria dos negócios são tidas como as principais vantagens competitivas do sistema de BI. Também foram indicadas como vantagens a destacar, o conhecimento empresarial, a competitividade e a flexibilidade. De acordo com os resultados obtidos da investigação, constata-se que um sistema de BI é capaz de oferecer uma maior eficácia no processo decisório na organização, e consequentemente melhora o próprio negócio.

Tendo por base essa pesquisa, pode-se afirmar resumidamente que a implementação e o uso do um Sistema de BI com sucesso numa determinada empresa ou organização, trará enormes vantagens à empresa, destacando-se nomeadamente a redução dos custos administrativos e corporativos, redução dos custos de avaliação de negócio, redução dos custos de avaliação da qualidade do serviço ou produto prestado, redução dos custos de avaliação do mercado onde está inserida a empresa, maior segurança e rapidez da informação para o processo de tomada de decisões estratégicas, maiores vantagens em relação à competitividade, melhoria na conceção dos relatórios administrativos, economia de tempo, processos mais eficientes, melhorias na qualidade de informação, ampliação da compreensão das tendências dos negócios, entre muitas outras vantagens que podem ser alcançadas de acordo com a natureza do negócio.

### 3.2.3. Sistemas transacionais (OLTP)

EVORA - HORTA DAS FIGUEIRAS  
Tel.: 266788240

Pingo Doce - Distribuição Alimentar, S.A.  
Sede: R Actor António Silva, N7, 1649-033 Lisboa  
Registo C.R.C. Lisboa-Matricula/NIPC: 500829993  
Registo Pingo Doce ANREEE: PT001730  
Capital Social: 33.808.115 EUR

Artigos		
TAKE-AWAY		
D 13% FRANGO ASSADO		4,36
FRUTAS E VEGETAIS		
C 6% MANGA PRONTA COME KG	0,686 X 4,49	3,08
	Poupança Imediata	(0,69)
C 6% TOMATE REDO 67/82 KG	0,330 X 1,99	0,66
C 6% SALADA CAMPONES 250G		1,49
Resumo		
TOTAL		9,99
TOTAL POUPANÇA		(0,69)
TOTAL A PAGAR		8,90

← Linha de produto

← Transação

Figura 3 - Exemplo de uma transação num talão de caixa de um supermercado

Segundo Antunes (2008, p. 2) *“Um sistema transaccional é o sistema básico de uma organização, que funcionando ao nível operacional, executa e regista as transações diárias efetuadas pela e na organização. (...) Os sistemas transaccionais são os sistemas fundamentais para o funcionamento diário de uma organização: qualquer falha num destes sistemas pode inviabilizar o normal funcionamento da atividade”*.

Tal como o nome indica são sistemas orientados para a transação, ou seja, trata-se de sistemas com a finalidade de registar as operações transaccionais diárias de uma empresa ou organização. Nos sistemas transaccionais, as informações estão bem estruturadas, elas asseguram a operacionalidade do negócio. Cite-se a título de exemplo o pagamento de um serviço de telecomunicação, ou a linha de um produto num talão numa compra efetuada num supermercado como observamos na Figura 3.

Este sistema, sendo considerado de vital importância hoje em dia, acaba mesmo por ser imprescindível para o funcionamento de uma empresa ou organização.

Caldeira (2012 p. 25) define um sistema transaccional como, *“ qualquer aplicação que realiza a gestão quotidiana das transaccões que explicam o modo como uma organização desenvolve as suas actividades.(...) as transações são usualmente efectuadas em tempo real. (...) são especialmente vocacionadas para o registo de dados”*. Resumidamente, podemos considerar que o OLTP é uma ferramenta de registo operacional diário, com o funcionamento típico de uma agenda, onde são escritas todas as operações transaccionais da empresa ou organização. É de realçar que os *“sistemas OLTP são fontes de dados que abastecem o data warehouse”* (Caldeira, 2012, p. 38).

### **3.2.4. Sistemas Analíticas (OLAP)**

Os Sistemas analíticos tiveram o seu aparecimento devido à necessidade de se poder dispor de ferramentas de análise e exploração dos dados contidos nos sistemas transaccionais. Por outras palavras, a OLAP é uma ferramenta de suporte à análise multidimensional, permitindo aos seus utilizadores observar as enormes quantidades de dados em diferentes dimensões, ou em diferentes perspetivas, oferecendo recursos potentes para análise dessas informações, de modo a obter-se uma maior perceção destes dados no processo decisório em qualquer ramo da empresa.

Segundo Caldeira (2012 p. 26) a aplicação OLAP *“está essencialmente interessada na determinação do melhor modelo para arrumar a informação de forma a maximizar a sua função analítica. (...) o objecto das aplicações OLAP não é a forma como a informação é guardada num repositório de dados, mas o modo como a mesma informação pode ser eficientemente analisada. (...) servem fundamentalmente para comparar grupos de dados relacionando-os com diversos critérios, de que se podem salientar a escala temporal”*.

### 3.3. OLAP versus OLTP

Quadro 1 -Características OLAP vs Características OLTP. Adaptado de (Elias, s.d.)

<i>Características</i>	<i>Sistemas OLAP</i>	<i>Sistemas OLTP</i>
<i>Armazenamento de dados</i>	O armazenamento é realizado em estruturas de data warehouse, com otimização do desempenho em grande volume de dados.	O armazenamento é feito em sistemas convencionais de banco de dados, através de sistema da informação da organização.
<i>Estrutura de dados</i>	Modelo dimensional. Normalmente, os dados possuem alto nível de sumarização.	Modelo relacional normalizado. Os dados possuem alto nível de detalhe.
<i>Finalidade dos dados</i>	Gerar informações confiáveis para suporte no processo decisório.	Registrar as operações transacionais diariamente.
<i>Fonte de dados</i>	Surgem de vários sistemas transacionais(OLTP).	OLPT são as fontes originais dos dados. Dados operacionais.
<i>Frequência de Atualização dos dados</i>	Baixa frequência. A atualização é feita por um critério específico. Ex (semanal, mensal).	Alta frequência. Atualização dos dados é feita no momento da transação.
<i>Granularidade</i>	Agregados	Atômico
<i>Orientação</i>	Arrays	Registos
<i>Quantidade de dados</i>	Muitos.	Poucos.
<i>Queries<sup>9</sup></i>	Queries relativamente simples, retornando poucos registros.	Uma vez que envolve agregação dos dados as Queries são complexas.
<i>Tipos de permissão de dados</i>	É permitido apenas inserção e a leitura. Para o utilizador é permitido somente a leitura.	Pode inserir, alterar, eliminar e atualizar.
<i>Tipos de Utilizadores</i>	Gestores e analistas, para tomadas de decisões.	Técnicos operacionais, engloba todos os utilizadores da organização.
<i>Utilizadores</i>	Poucos	Muitos
<i>Volatilidade</i>	Dados históricos e não voláteis. Os dados não sofrem alteração. Exceto por motivos necessários. Ex (motivos de erros).	Dados atuais e voláteis, passíveis de modificação e exclusão.

<sup>9</sup> Consultas em Português.

### 3.4. Ferramentas de Business Intelligence

A arquitetura pode ser considerada como um guia para os responsáveis da implementação, desenvolvimento e manutenção de um sistema de BI. Normalmente, um sistema típico de BI é composto pelas seguintes partes: Fontes de dados, ETL, Data warehouse e Análise de dados, como é demonstrado na Figura 4. Para um boa implementação e funcionamento, é necessário que a arquitetura do sistema não exclua nenhum desses passos.

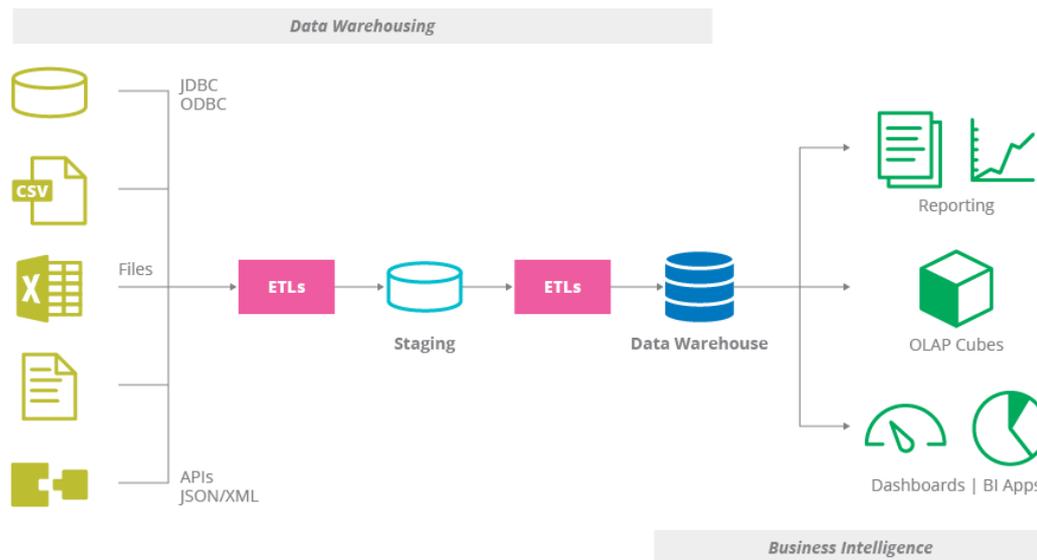


Figura 4 - Arquitetura de apoio ao processo de BI. Fonte: (Murillo, 2016)

Na Figura 4 é apresentado, de forma simplificada e compreensível, cada componente integrante da arquitetura de um processo de sistema de Business Intelligence. Podemos observar várias etapas imprescindíveis a seguir para implementação desse sistema. Considerando a abordagem desta arquitetura, podemos destacar as seguintes fases para implementação de um sistema de BI:

- **Fonte de dados**, é onde se encontram todas as origens dos dados de que necessitamos para extração dos dados, se for necessário realizar o seu tratamento; em seguida efetuar o carregamento no sistema. Na Figura 4 observam-se as seguintes fontes: base de dados relacionais, arquivos do Excel, ficheiros CSV, entre outras. De acordo com Caldeira (2012 p. 38), “as fontes podem ser internas ou externas. As internas são provavelmente as mais frequentes e são aquelas com que a organização lida no seu dia-a-dia como compras, vendas processamento de vencimento” etc.(...) As fontes externas são fontes de informação que no ambiente de exploração transaccional não existem, porque não fazem sentido naquele momento”. A título de exemplo podemos mencionar, um simples ficheiro com os indicativos telefónicos de cada país, ou todos os codigos postais de um determinado país.
- **ETL**, o processo de extração, tratamento e carregamento de dados, por outras palavras resume-se em Área de Estágio. Tem como finalidade a utilização das ferramentas apropriadas para extração dos dados proveniente das diversas

fontes, colocação num repositório de dados temporário, realização e tratamento dos mesmos, e finalmente carregamento dos dados para o repositório de dados central (DW). Segundo Caldeira (2012, p. 39), este “*processo engloba todas as operações relacionadas com a localização dos dados, rotinas de extração, escolha, limpeza, transformação e carregamento da informação transacional no data warehouse*”.

- **Data Warehouse (DW)**, é um repositório de dados organizacional que é a base principal para as atividades de análise de dados. Em outras palavras deve ser o espelho do negócio, com o objetivo de organizar informações de modo multidimensional, facilitando a exploração dessas informações. Segundo Caldeira (2012, p. 16), “*a data warehouse tem essencialmente a ver com a autonomia do utilizador, ou seja, o utilizador é um agente ativo na exploração dos dados*”. Ela não é um sistema para registar dados; ela é uma imagem do sistema OLAP de uma organização ou empresa, num dado momento de tempo.
- **Análise de dados (OLAP)**, após essas três camadas, finalmente temos a camada final da arquitetura que consiste na exploração dos dados e armazenamento no repositório de dados central (DW). De forma a melhorar a performance de análise de dados, é aconselhável aplicar as técnicas de Business Intelligence. Um dos objetivos do nosso estudo consiste em utilizar essas técnicas na rede prosopográfica, para aprimorar a análise de dados. Acedendo ao DW, nomeadamente com a técnica OLAP, podemos gerar informações relevantes para o processo decisório, que são apresentadas em aplicações de Front-End, utilizando gráficos, mapas, dashboards entre outras.

Resumidamente, com as ferramentas de Business intelligence, extraímos em primeiro lugar os dados das diversas fontes, e em seguida realizamos o seu tratamento e carregamento, analisamos os dados contextualizados, procurando estabelecer relações entre um conjunto de dados, e transformamos os registos obtidos em informações úteis que resultam do processamento de um conjunto de dados que possuem uma diversificação de atributos importantes como data, lugar, forma, entre outras, para o conhecimento empresarial.

### 3.5. Processo de Extração Transformação e Carregamento (ETL)



Figura 5 - Modelo da área de extração, transformação e carregamento do DW

Citando Caldeira (2012, p. 205), o processo de ETL é entendido como “*subsistema de extração, transformação e limpeza dos dados dos sistemas operacionais no momento da sua disponibilização no data warehouse.*” O mesmo autor refere que a implementação de um sistema de BI, é um dos processos com maior complexidade, e durabilidade, podendo ocupar até 80% do trabalho. Tem como objetivo primordial, a extração dos dados nos sistemas transacionais, transportando-os para a área de estágio, onde ocorrerá a sua transformação e posteriormente o seu carregamento no DW (Caldeira, 2012, p. 39).

A Figura 5 é representativa do processo de um sistema ETL, onde são processadas enormes quantidades de dados, tornando-se por esse motivo, um processo mais crítico na construção de uma DW. Face ao exposto, é fundamental a escolha de ferramentas adequadas para este processo. Ferreira, Miranda, Abelha e Machado (2010, p. 5-6), elencam os seguintes pontos a levar em consideração na seleção da ferramenta:

- Deve suportar qualquer plataforma;
- Dispor de uma usabilidade simples;
- Ter a eficácia e eficiência de efetuar a leitura dos dados diretamente da fonte de origem, em diferentes formatos;
- A capacidade de extração dos dados nas diversas fontes, efetivar a sua limpeza, transformação, agregação e carregamento no DW;
- Ter suporte para programação em linha de comandos usando programação externa;
- Apoiar a reutilização da lógica de transformação para que o utilizador não necessite de reescrever, várias vezes, a mesma lógica de transformação;
- Apoio a nível do debugging, ou seja, deve oferecer apoio de execução e a limpeza da lógica de transformação. O utilizador deve ser capaz de visualizar os dados antes e depois da transformação;
- Rapidez e flexibilidade.

O mercado dispõe de inúmeras ferramentas de ETL com características diferentes, podendo indicar-se a título de exemplo algumas que se destacam pelas suas capacidades de tratamento, manipulação de forma simples e eficaz: o OWB, Data Stage<sup>10</sup>, SSIS<sup>11</sup>, as open source como Talend e PDI ou Kettle<sup>12</sup>.

A primeira e segunda fase do processo ETL, necessita de uma atenção especial. Os dados extraídos e transformados, precisam de gerar informações confiáveis, caso contrário, as decisões serão baseadas em informações erróneas, podendo assim afetar diretamente a sobrevivência do negócio. Na fase de extração temos que ter a precaução de não alterar os dados na fonte de origem. Segundo Rainardi (2008, p. 173) a fase de transformação dos dados, resume-se à padronização, formatação e agregação dos dados, ou seja, refere-se à compactação dos dados de nível mais elevado, e a realização das limpezas dos dados no

---

<sup>10</sup> **Data Stage** - IBM Information Server, foi desenvolvida pela Ascential Software, adquirida pela IBM em 2005. É uma das ferramentas ETL mais conhecidas e utilizadas no mercado mundial.

<sup>11</sup> **SSIS** – Interface para integração de dados, estruturado no Visual Studio.

<sup>12</sup> **SPOON** é a ferramenta gráfica, para desenhar e testar todo o processo do PDI.

formato de texto, números; há que ter sobretudo atenção especial no formato das datas, de acordo com a natureza do projeto envolvido. Há fontes de dados que não requerem muita transformação dos dados, podendo em determinados casos ser imprescindível proceder a muitas limpezas e manipulação desses dados.

Por exemplo, não queremos que um número de telefone seja inválido, ou um endereço de email sem o «@», ou ainda um endereço de Évora com um código postal não correspondente a esse endereço, diversas formas de escrever nome cidade, a título de exemplo «Entre Douro e Minho», «Entre-Douro-Minho», uma coordenada geográfica que não corresponda à localização correta, diversos modos de representar o sexo, por exemplo «Masculino e Feminino», «M ou F», «1 para Masculino, 0 para Feminino».

Constituem os exemplos supracitados tipologias de modificações, a que deveremos estar atentos e que devem ser verificados na fase de transformação, em seguida efetuar-se a sua limpeza, de forma a obtermos dados com boa qualidade, integridade e fiabilidade.

Após a extração, transformação e limpeza dos dados, passamos à fase de carregamento que consiste em colocar os dados no DW. Este processo varia de acordo com a necessidade do projeto. Por exemplo em alguns projetos os dados são carregados e atualizados semanalmente ou diariamente, enquanto que noutros casos são adicionados a cada hora, até em cada transação. Deste modo, o carregamento dos dados depende da necessidade do negócio em ter as informações disponíveis para suporte nas tomadas de decisão. A título de exemplo podemos mencionar o processo ETL de um sistema de BI de um supermercado, que é diferente do sistema SPARES; há uma maior necessidade de carregamento de dados no sistema de BI do supermercado, pela simples razão de que a regra de negócio exige tomar as decisões com maior regularidade, do que no sistema SPARES.

Segundo Rainardi (2008, p. 176) podemos usar as seguintes abordagens para o processo de ETL:

- A abordagem mais comum é o processo ETL poder consultar as fontes de dados de origens regularmente, e efetuar a extração, transformação e carregamento dos dados;
- Outra abordagem é através dos Triggers, que vão estar associados a um evento. No momento em que o evento ocorrer, o trigger é acionado, sendo responsável pela inserção ou atualização dos dados numa determinada dimensão. É esta abordagem que utilizamos para o sistema SPARES. Na tabela do facto “Comissario”, quando um utilizador inserir um novo registo, o trigger associado para essa função, vai atualizar os novos campos («Código da Data Inicial,» «Código da Data Final», «Ocupações») adicionando na tabela de facto, sendo desta forma que será o processo ETL para o sistema SPARES;
- Uma outra abordagem menos utilizável, é através de um leitor de Log, onde se efetua a leitura de Log da base de dados para identificar alguma alteração nos dados, e em seguida obtém-se os dados modificados e armazena na DW.

Resumidamente, as três letras são de fácil compreensão: a letra (**E**) consiste em extrair os dados da fonte de origem, a letra (**T**), representa a transformação dos dados, e a letra (**L**) a posteriori, efetua o carregamento, em um conjunto de dimensões emparelhados numa tabela de facto. No processo ETL a fase de extração e carregamento é obrigatória, sendo que, a transformação e a limpeza dos dados são opcionais. Concluindo, o processo ETL deve ser robusto, e em caso de falhas deve oferecer a recuperabilidade sem danos ou perdas dos dados.

### 3.6. Dashboards e Scorecards

#### 3.6.1. Conceito de Dashboards e Scorecards

Nenhum gestor é capaz de olhar e examinar milhares de dados, centenas de colunas a «*olho nu*» sem um sistema de apoio, e retirar informações úteis para uma decisão que até pode ser vital para a falência ou sobrevivência da sua empresa. É nesse contexto e problemática que lhe está associada, que surgem os dashboards e scorecards.

*“Um scorecard ou dashboards deve funcionar em simbiose com o cérebro humano. O cérebro funciona por associação, por isso tantas pessoas têm as suas mnemónicas preferidas. O cérebro humano gosta de testar/avaliar hipótese, de prever o que poderá obter sob determinadas condições. Ora isto é precisamente o mundo da análise de dados.”* (Caldeira, 2012, p. 51). Citando o mesmo autor, quando as dashboards são suportadas por uma data warehouse, pode usufruir-se do máximo das suas potencialidades, dado que o data warehouse é desenvolvido com o propósito de simplificar a análise e a exploração dos dados. Isto não configura a não utilização de outras quaisquer estruturas de dados, contudo, os resultados obtidos não têm a mesma qualidade quando o sustentáculo é data warehousing.

Pode-se designar dashboards ou scorecards como sendo painéis de controlo e acesso a um conjunto de informações e indicadores pertinentes para determinada atividade na organização. Executa a comunicação de uma enorme quantidade de dados, de modo sintético, através do uso de recursos gráficos; esses dados devem ser concisos, claros e intuitivos, de maneira a que a transmissão das informações seja da melhor forma possível, executando-se a mesma em menor tempo possível (Costa, 2014, p. 7).

Na atualidade, as organizações e as empresas para acederem às informações, preferem os dashboards, devido ao facto de apresentarem visualmente as informações de forma atraente e apelativa. As informações podem ser mostradas através de tabelas dinâmicas, gráficos de dispersão e de área, manómetros, diferentes tipos de mapas, barras horizontais, exibições circulares, histogramas, possibilitando-se a visualização interativa, e a possibilidade de manuseamento por intermedio das operações de *drill* e *slice and slice*.

Podemos considerar dashboards como sendo, uma ferramenta fundamental de qualquer organização, adequada ao acompanhamento diário dos seus indicadores. Com o objetivo primordial de apoiar os gestores no processo decisório, permitirá o alerta sobre o desenvolvimento do negócio, e no caso de haver algum desvio ou ameaça, servirá como suporte na identificação e diagnóstico das causas das ameaças, e permitirá reforçar o planeamento estratégico da organização.

### 3.6.2. As Finalidades de Dashboards e Scorecards

Dashboards é uma ferramenta que proporciona a monitorização em conjunto de um vasto número de informações relevantes, num único ambiente. Segundo Costa (2014, p. 6-11), ela tem como principais finalidades:

- Exibir as medidas e indicadores preponderantes do desempenho da organização, e as medidas cruciais para as tomadas de decisões estratégicas, através de gráficos, tabelas, outras formas;
- Proporcionar aos gestores o acesso contínuo ao desempenho da organização;
- Disponibilizar informação de forma correta, atual e confiável;
- Conceder ao utilizador a interatividade por níveis de detalhe distintos, permitindo filtrar os dados apresentados, de acordo com as necessidades evidenciadas pelo mesmo;
- Oferecer formas de visualização onde transpareçam as prioridades, metas e os níveis de desempenho desejados pela organização, atividade ou negócio;
- Oferecer opções de alerta para oportunidades e riscos integrados com os demais sistemas de comunicação organizacionais;
- Permitir a integração com fontes externas de dados;
- Oferecer formatos variados de documentação e circulação da informação proveniente da ferramenta.

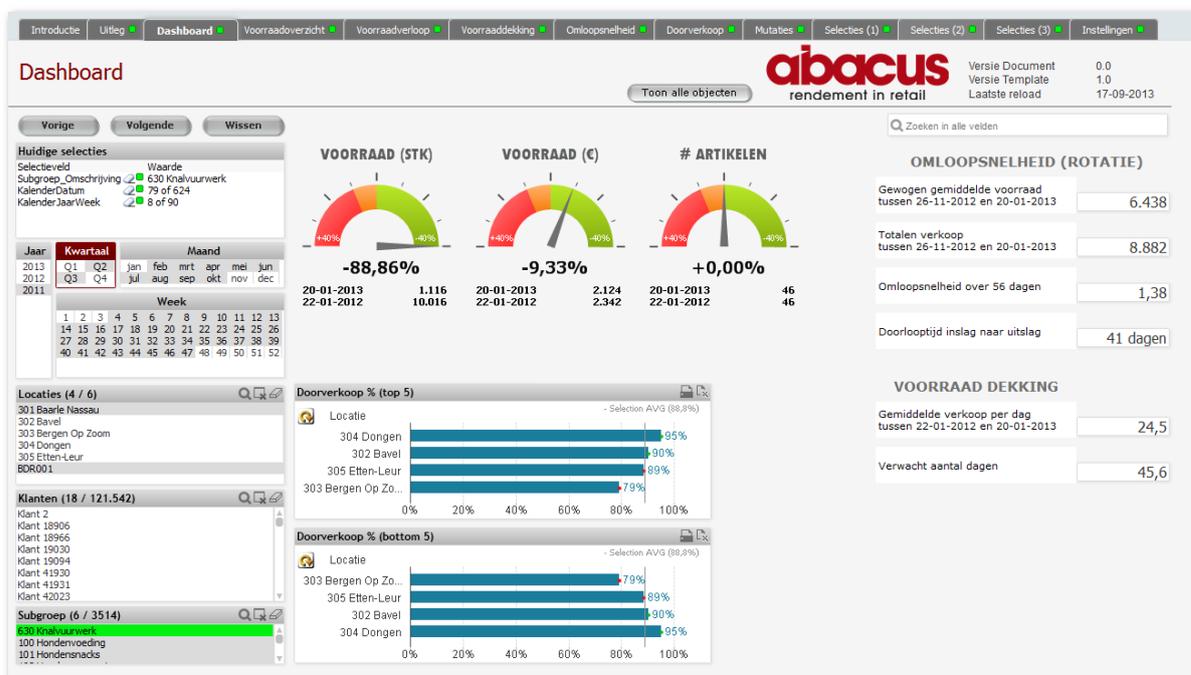
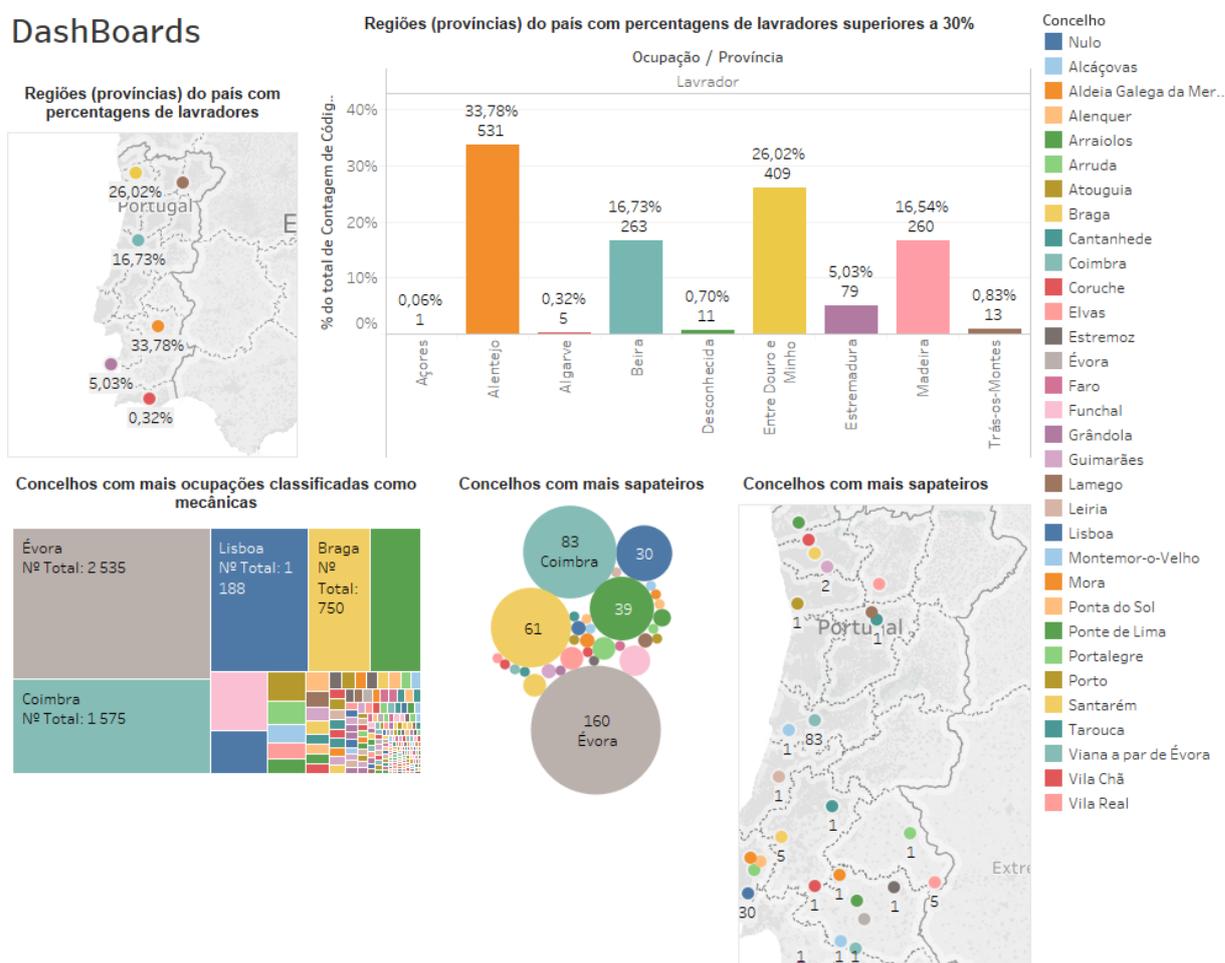


Figura 6 - Resultados obtidos em Dashboards. Fonte: (Abacus, s.d)

Assim, os dashboards distinguem-se das demais ferramentas de análise e exploração dos dados, principalmente pelo facto de oferecerem uma visão global de todas as informações e estabelecerem a filtragem para eliminar as informações irrelevantes do momento.

Segundo Caldeira (2012, p. 51), “o dashboarding é um método rápido e eficiente de captura e análise de dados, e a ferramenta ideal de exploração da informação num data warehouse”, sendo por esta razão, que o objetivo final do estudo para além dos dados serem apresentados no ecrã, é construir um protótipo constituído por um conjunto de indicadores consubstanciados em dashboards, de acordo com o representado na Figura 6. Apriori os dashboards obtidos podem ser exportados para PDF, documento de texto, apresentação de slides, entre outros diversos tipos de formatos.



### 3.7. Cubos Multidimensionais (OLAP)

Hoje em dia, o armazenamento de dados numa empresa ou organização, é uma necessidade inevitável. Contudo, não basta ter os dados armazenados, sem que a posteriori possam ser usados, para nos dar informações valiosas, sendo até determinante para a sobrevivência do processo de negócio. Na atualidade, para os profissionais que ambicionam entender, aprender, analisar ou exercer um trabalho ou projeto relacionado com a área de inteligência dos negócios, é imprescindível, o conhecimento acerca das técnicas de construção de cubos de dados analíticos ou multidimensionais.

A nível de contextualização histórica, foi nos finais da década de 90, que se verificou o aparecimento dos cubos analíticos (OLAP). Atualmente, a Microsoft, IBM e Oracle são as empresas mais conhecidas por produzirem ferramentas para OLAP. Segundo Codd, Codd, e Salley (1993, p. 12) este termo foi usado pela primeira vez por Edgar Frank Codd, que definiu as seguintes regras fundamentais para um aplicativo do tipo cubo, e que infra se elencam:

1. **Dimensionalidade genérica** – cada dimensão deve ser equivalente na sua estrutura e capacidade de operacionalidade;
2. **Suporte a multiutilizadores** - as ferramentas OLAP devem fornecer acesso simultâneo à recuperação e atualização, integridade e segurança;
3. **Conceito de visão multidimensional** – a visão multidimensional dos dados permite, uma manipulação simplificada e intuitiva dos dados, uma vez que inclui a ferramenta “*Slice and dice*<sup>13</sup>”;
4. **Transparência** – a aplicação do tipo cubo tem que ser apresentada de forma transparente para o utilizador final, porque ela é crucial para preservar a produtividade de informações de boa qualidade;
5. **Acessibilidade** - o utilizador final não deve ter nenhuma preocupação com a fonte de dados, devendo pelo contrário a aplicação OLAP aceder e disponibilizar, somente os dados necessários para a análise indicada;
6. **Desempenho consistente de relatórios** - o desempenho da ferramenta OLAP não deve ser afetado significativamente à medida que o número de dimensões é aumentado;
7. **Arquitetura cliente/servidor** - o servidor da ferramenta OLAP deve ter a capacidade de poder ser usado por vários utilizadores. O servidor deve ser capaz de mapear os dados de diferentes bases de dados;
8. **Tratamento dinâmico de matriz esparsa** - a estrutura física servidores OLAP deve ter um tratamento ótimo para matriz esparsa. Considera-se matriz esparsa aquela em que nem todas as células na matriz contêm dados. Quando confrontado com uma matriz esparsa, o sistema OLAP deve ser capaz de deduzir a distribuição dos dados e saber como armazená-la da forma mais eficientemente possível;
9. **Operações de cruzamento dimensional irrestritas** - na análise de dados multidimensional, todas as dimensões são criadas e tratadas de igual modo. Por exemplo, um utilizador executa as mesmas ações sobre as dimensões Data e

<sup>13</sup> **Slice and dice** - funciona como uma espécie de relatório. No cubo pode ser criado através de arrastar uma ou varias dimensões e seus atributos para a visualização dos seus dados. Podemos afirmar que forma um “sub cubo”.

localidade. As ferramentas OLAP devem lidar com os cálculos nas associações entre as dimensões e não exigir que o utilizador defina, a forma de efetuar esse cálculo;

10. **Manipulação de dados intuitiva** – qualquer manipulação de dados inerente ao caminho de consolidação de dados deve ser realizada através de uma ação direta nas células do modelo analítico, e não devem exigir o uso de um menu ou navegação pela interface do utilizador;
11. **Relatórios flexíveis** - a função de gerar relatórios deve poder apresentar os dados a serem sintetizados ou as informações resultantes da animação do modelo de dados em todas as orientações possíveis;
12. **Níveis de dimensões e agregações ilimitados** - a recomendação é que uma ferramenta OLAP possa acomodar varias dimensões, num modelo analítico e agregações de factos ilimitados.

No enquadramento da construção de uma aplicação do tipo CUBO, verifica-se que o que se apresenta ao utilizador é a informação numa visão cúbica. Conforme analogicamente podemos observar na Figura 8, trata-se do output dos servidores OLAP, em que as dimensões e factos da data warehousing, são diretamente mapeados para esta aplicação, permitindo ao utilizador mudar as suas arestas e segmentos, de acordo com as suas necessidades de pesquisa de dados.

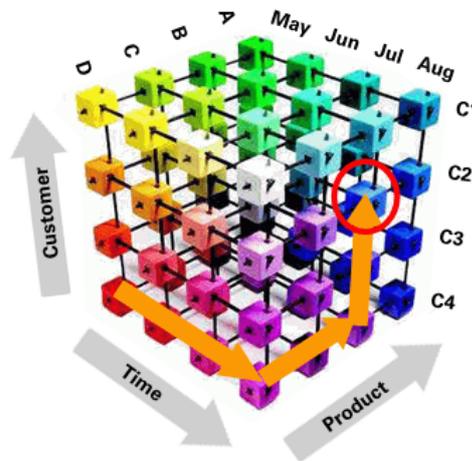


Figura 8 - Aplicação do tipo Cubo. Fonte: (Oracle, s.d).

De acordo com Kimball e Ross (2013, p. 9) na construção de um aplicativo tipo cubo analítico, é necessário que os dados para exploração, tenham o suporte em **Star schema**<sup>14</sup>( é explicado no capítulo 4.8 ), ou seja, um esquema constituído por uma tabela de facto, que está associada a várias dimensões. Devido à sua simplicidade, esta tabela facilita a compreensão do funcionamento do processo de negócio aos investigadores. É um passo indispensável na construção de um cubo, uma vez que ela constitui a base estável no desenvolvimento e na compreensão do aplicativo tipo cubo.

<sup>14</sup> **Star Schema** – Esquema em Estrela em português.

Através da construção de um Cubo semelhante ao da Figura 9, os utilizadores do sistema SPARES, vão poder usufruir de uma nova ferramenta de análise de dados, poderosa e dinâmica, que lhes permitirá manipular os dados, e terem à sua disposição o poder de alterarem as arestas e os segmentos do cubo de acordo com as suas necessidades. Após obter os resultados no ecrã de análise de dados, pretende-se com os APIs de google maps imprimir os resultados em mapas nos casos possíveis.

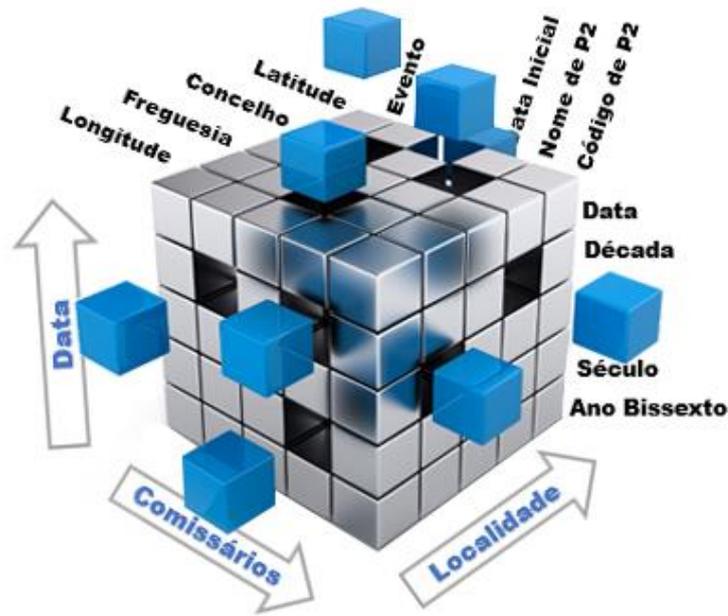


Figura 9 - Aplicação do tipo Cubo para o SPARES

De modo a familiarizarmo-nos com a arquitetura do cubo multidimensional, é fundamental que se compreendam algumas expressões relacionadas com a mesma, entre as quais se destacam:

- **Dimensão** - observando a Figura 9, podemos defini-la como sendo uma face do cubo, ou por outras palavras uma unidade onde se agrupam determinados atributos de negócios, e que constitui a base para a análise de dados. No sistema SPARES podemos mencionar «localidade», «data», «ocupação» como exemplos;
- **Membro** - é um subconjunto de uma dimensão;
- **Medidas** - são as parametrizações usadas no processo decisório, sendo conotadas normalmente com dados numéricos que permitem realizar comparações, podendo-se mencionar a título de exemplo as «quantidades» e «médias»;
- **Hierarquia** - é a estruturação em árvore, tendo como função primordial, a organização dos membros de uma determinada dimensão;
- **Nível** - dentro de uma hierarquia, as informações podem estar estruturadas de acordo com a granularidade, ou seja, com os seus níveis de detalhe, como por exemplo na dimensão Data em que há o Ano, Semestre, Trimestre, Mês, Semana e Dia.

### 3.7.1. Tipos de Operações no Cubo Multidimensional.

Segundo Antunes (2008, p. 16) a aplicação tipo cubo, é uma das estruturas de dados, com maior usabilidade para manipular os dados que alocam um determinado modelo multidimensional. “As arestas do cubo correspondem às diferentes dimensões do modelo, e a intersecção dos valores de cada dimensão, designadas células, expressam a agregação dos factos registados na combinação dessas condições”. O Quadro 2 abaixo indicado, elenca os tipos de operações que permitem a manipulação do cubo.

Quadro 2 - Tipos de operações no Cubo. Adaptado de: (Ciferri e Ciferri, s.d)

<i>Tipos de Operações</i>	<i>Definições</i>
<b><i>Drill-Down</i></b>	Desagrega uma dimensão, ou por outras palavras, analisa os dados em níveis de agregação progressivamente mais detalhados, ou de menor granularidade.
<b><i>Drill Across</i></b>	Compara medidas numéricas distintas, que são relacionadas entre si, através de pelo menos uma dimensão em comum.
<b><i>Pivot</i></b>	Tem o objetivo de modificar o eixo de visualização, ou seja, reorientar a visão multidimensional, oferecendo diferentes perspetivas dos dados.
<b><i>Roll-Up</i></b>	Sumarizar dados n por subida na hierarquia ou por redução de uma dimensão n, resumidamente significa a agregação de uma dimensão.
<b><i>Slice and dice</i></b>	Restringe os dados, sendo analisados um subconjunto destes dados slice: corte para um valor fixo dice: seleção de faixas de valores.

A aplicação de tipo Cubo Multidimensional, escolhido para as explorações e análise de dados no sistema SPARES, é o **Tableau**.

De acordo com a Figura 10, no corrente ano, o Tableau foi reconhecido como líder no Quadrante Mágico Gartner para plataformas de Business Intelligence. Nos últimos cinco anos, as maiores empresas do mundo, entre diversas ferramentas de BI como a Power BI da Microsoft, Qlik, SAS, IBM, têm optado pelo Tableau, por algumas das razões abaixo transcritas, e que, face à fiabilidade das mesmas, nos levou a escolhê-lo para analisar os dados no sistema SPARES.

- **Aspeto visual**, é uma das principais características que leva as pessoas a escolherem o Tableau, devido à sua simplicidade de visualização e utilização.
- **Inovação**, a introdução de VizQL, o processamento e pesquisa em linguagem natural, são das maiores inovações a serem introduzidos no Tableau.

- **Suporte a várias Plataformas**, hoje em dia suporta os dados nas nuvens, no localhost, e brevemente vai suportar também no Linux.
- **Maior impacto no Negócio**, é uma ferramenta que quando for implementada permitirá aos gestores usufruir de todas as suas funcionalidades, devido à sua fácil usabilidade.
- **Licença gratuita para Estudantes e Professores**, essa é uma das razões mais importantes de termos escolhido o Tableau, uma vez que fornece o Serial Key gratuito para todos os estudantes e professores. Dado que todos os utilizadores do sistema SPARES, são professores ou funcionários da Universidade de Évora, têm direito a chave de ativação do Tableau, com todas as funcionalidades.



**Figura 10 - Magic quadrant for BI and analytics platforms.**

Fonte: (Gibson, 2017)

Em suma, neste capítulo, referenciamos as operações fundamentais para exploração de dados multidimensionais, disponíveis em cubos. Estas operações, manipulam o cubo, sobretudo as operações Roll-Up e Drill-Down que aumentam ou diminuem o nível de granularidade, enquanto que as operações Slice e Dice aplicam filtros sobre o mesmo cubo.

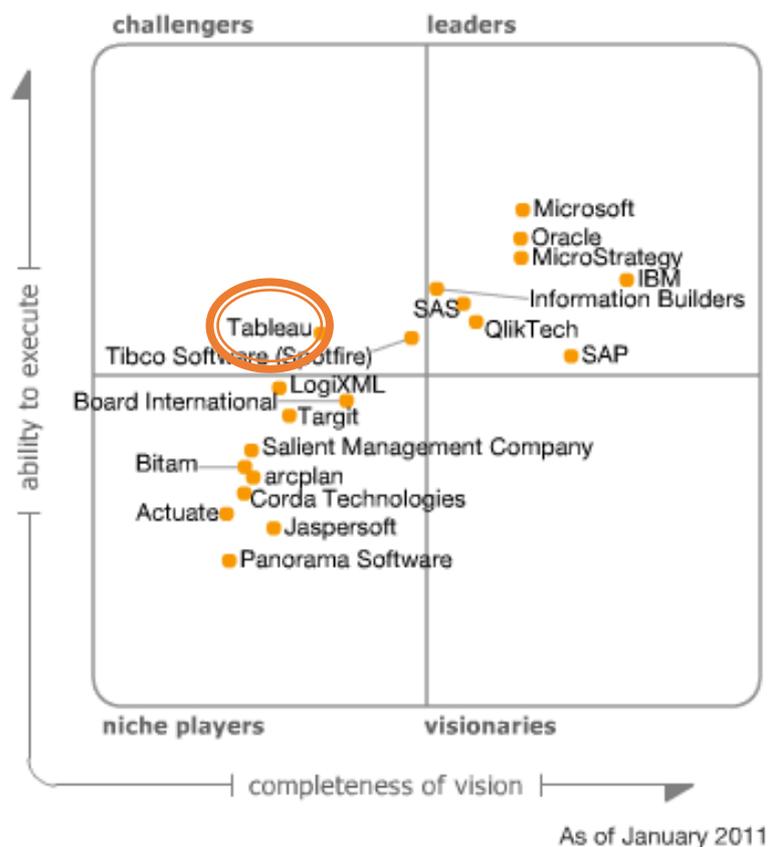


Figura 11 - Magic quadrant for BI and analytics platforms.

Fonte: (Huijbers, 2011)

Como podemos constatar na Figura 11 o Tableau há sensivelmente 5 anos atrás, não tinha destaque no quadro mágico de Business Intelligence. E se remontarmos ao ano de 2010, constatamos que o Tableau nem sequer aparece, mas a Figura 10 mostra-nos os dados mais atuais em que o Tableau surge como o líder do mercado de Business Intelligence.

### 3.8. Análise e Exploração dos Dados

Atualmente, existem diversas organizações que a nível da exploração de dados, componente essencial para a tomada de decisões efetivas e assertivas, ainda utilizam os sistemas transacionais, que não permitem a obtenção de uma resposta satisfatória face às exigências organizacionais. Uma das razões para que tal ainda se verifique, está relacionada com a incapacidade daquelas organizações manipularem de forma eficiente dados agregados, capazes de explicitarem as regras do negócio.

Geralmente, as pesquisas aplicadas a uma ou mais tabelas do sistema transacional, constituem uma operação de alto custo, nomeadamente em relação à velocidade em obter os resultados da análise realizada, uma vez que o método utilizado são os **Joins**, que é a forma de relacionar as tabelas no sistema OLTP, e normalmente exigem algum conhecimento informático.

No sistema SPARES o único método de realizar as pesquisas, é através de filtros. Entre outras lacunas, elas não possibilitam pesquisas de relacionamento de dados de duas tabelas. De forma a simplificar as explorações dos dados, oferecendo outra alternativa eficaz e eficiente, surge a aplicação do tipo cubo multidimensional dando alternativas aos investigadores na exploração dos dados no sistema SPARES.

Essa aplicação tem mecanismos de cálculo, que a partir de factos com um grau elevado de atomicidade produz um resultado sumariado, através do método de agregação, tornando assim as análises mais eficientes. Segundo Antunes (2008, p. 43), este aumento de eficiência deve-se sobretudo, à possibilidade de armazenar de modo persistente, os dados agregados sob a forma de cubos multidimensionais, suprimindo-se a necessidade de calcular as agregações durante a fase de análise.

Em concreto, a aplicação tipo cubo implementado para análise de dados no sistema SPARES, distingue-se da forma atual (Filtros) de análises dos dados, essencialmente por várias razões, mas em especial destacam-se duas vertentes: na manipulação e agregação dos dados, e na sua finalidade em descobrir novos padrões de dados, que não é possível com os filtros.

Deste modo, o sistema de analisar os dados através do cubo multidimensional, pretende oferecer um acesso rápido e flexível, permitindo para tal a identificação de tendências temporais e outras relações importantes relacionadas com a atuação dos comissários no Santo Ofício.

Pretende-se como objetivo central, que essa aplicação para análise de dados, se possa tornar uma ferramenta de trabalho amplamente usada e aceite pela CIDEHUS, reconhecendo-se as suas potencialidades, o seu alto grau de desempenho, a sua eficiência e eficácia na exploração de dados, bem como a sua facilidade na elaboração de relatórios, e análise de dados no repositório de dados prosopográfica.

# Capítulo 4

## **O Modelo Dimensional**



## 4. O Modelo Dimensional

A construção de uma data warehouse pode considerar-se, que assenta no modelo dimensional. A simplicidade do modelo dimensiona, facilita a compreensão dos dados e a sua análise, para obtenção de informações de boa qualidade. Estabelecendo um paralelismo analógico, podemos verificar que do mesmo modo que na arquitetura de uma casa, por mais simples que seja, é elementar e essencial ter-se um bom alicerce, também na conceção de um modelo dimensional, é fundamental ter-se uma boa base de suporte, que neste caso seria a determinação do processo de negócio, nomeadamente a construção da matriz em bus, não esquecendo de definir a granularidade mais atómica possível, para que possa proporcionar a agregação dos dados.

O modelo relacional devido à sua alta normalização, é usado comumente para tarefas de inserir, apagar e atualizar dados, enquanto que o modelo dimensional é diferenciado por ser demasiadamente desnormalizado, devido à sua base de suporte ser um esquema em estrela, onde todas as dimensões estão ligadas a uma tabela de facto central. Deste modo, ao contrário do modelo relacional, não há nenhuma preocupação acrescida com a redundância de dados.

Segundo Kimball (2002, citado por Barbosa e Oliveira (2007, p. 36)), o modelo dimensional é dirigido aos problemas de esquemas demasiadamente complexos na área de apresentação dos resultados. Dessa forma, não constitui uma cópia do modelo relacional, embora contenha os mesmos dados que um modelo normalizado. Entretanto, a estruturação dos seus dados, tem um formato diferente do sistema normalizado, devido ao seu objetivo primordial ser o de atender às necessidades dos utilizadores, no desempenho da análise de dados.

O modelo dimensional é formado por ter elementos essenciais: a Tabela de Facto, as Dimensões e as Medidas. Tem como objetivo de primeira importância, apresentar aos utilizadores os dados de uma forma transparente e com múltiplas utilizações.

No capítulo 4.5 descreve-se a tabela de Facto do sistema SPARES, acompanhado das medidas, e no capítulo 5.1.2 enunciam-se detalhadamente as dimensões que compõem o sistema SPARES. O conjunto das medidas que expõem os aspetos do negócio, estabelecem o modelo dimensional, que permitirá realizar a sumarização periódica ou transacional, garantindo deste modo, a estruturação, hierarquização e agregação dos dados, oferecendo igualmente um grande suporte à análise de dados para o processo decisório.

Mesmo tendo uma grande aceitação, existem algumas perceções erradas acerca do modelo dimensional. Segundo Kimball e Ross (2013, p. 31) os maiores mitos acerca do modelo são as seguintes:

- Modelos dimensionais e data marts são somente para dados sumarizados;
- Modelos dimensionais e data marts são soluções departamentais, não empresariais;
- Modelos dimensionais e data marts não são escaláveis;

- Modelos dimensionais e data marts são apropriados somente quando há um padrão de uso previsível;
- Modelos dimensionais e data marts não podem ser integrados e levam a soluções isoladas.

#### 4.1. Modelo Inmon Versus Modelo Kimball

Para a implementação de uma data warehouse, existe uma panóplia de ferramentas, de livre acesso ou comerciais, para o seu desenvolvimento, armazenamento e manutenção. Todavia, nesse leque de variedade de escolha das ferramentas a utilizar, podemos encontrar um elo comum, que é a utilização da modelagem centrada, no Modelo de Inmon ou Modelo de Kimball.

Segundo Gomes (2010, p. 9) escolher entre esses dois modelos, “*corresponde a uma escolha a nível de arquitectura e metodologia*”. Compreender os fundamentos da arquitetura e da metodologia desses dois modelos, proporciona um conhecimento basilar do conceito e funcionamento de DW.

Bill Inmon considerado o «Pai da Data Warehouse», em 1990 cunhou esse termo, na publicação *Building the Data Warehouse*, e desde então, as organizações introduziram a implementação de DW, segundo a visão de Inmon, e o grau de sucesso foi diversificado. Depois da publicação do livro de Inmon, outros especialistas de base de dados, iniciaram a investigação e implementação da data warehouse.

Nomeadamente, Ralph Kimball devido à sua experiência, reconduziu-o ao desenvolvimento de uma metodologia própria tendo, em 1998, publicado *The Data Warehouse Toolkit*. A posteriori Kimball publicou uma segunda edição da sua obra em 2002, recomendando nesta versão uma arquitectura de múltiplas Data Marts organizadas por áreas de negócio. Nesta versão, o DW é definido como sendo a soma dos vários Data Marts (Gomes, 2011, p. 10).

Os pontos comuns entre os dois, é considerarem que a solução completa é muito complexa para ser realizada de uma só vez, a necessidade de uma arquitetura, o esquema em estrela é de grande importância para apresentar as informações aos utilizadores, e também é importante o sistema de Metadados.

As suas ideias divergem em alguns pontos da arquitetura (ODS, DM), nomeadamente, qual a modelagem a utilizar (normalizado ou desnormalizado), e discordam também no papel dos Data Marts e na abordagem de projeto e construção do DW.

A principal diferença de abordagem entre o modelo de Kimball e de Inmon, é não tanto a nível conceptual, mas sobretudo a nível da terminologia usada. Inmon refere-se ao DM como uma coleção de dados derivado de DW, enquanto que Kimball alega que a DM é a própria unidade lógica do DW.

#### 4.1.1. O Modelo Inmon

De acordo com Gomes (2011, p. 15), Inmon vê o data warehouse como uma parte integrante de todo um sistema, o que significa que o DW e as BDs transaccionais são parte de um todo maior. *“A aproximação evolutiva de Inmon tem por base a tecnologia de base de dados relacionais, os seus métodos de desenvolvimento o design de sistemas transacionais, assentando em princípios e práticas usadas nas décadas anteriores. Podemos assim afirmar neste contexto que o modelo de Inmon surge como uma evolução natural do modelo relacional”*.

Para Inmon, Data Warehouse e Data Marts são estruturas essencialmente distintas. Defende que é difícil integrar um conjunto de DM, de forma a gerar uma DW completa. Sabendo que um DM deriva de uma DW, aconselha a implementação iterativa de DW, ou seja, o sistema inicia pequeno, evoluindo paulatinamente em espaço curto de tempo.

Esse modelo tem sucesso nas grandes organizações com muitas unidades de negócio diferentes que precisam compartilhar informações. Tem falhas devido à pouca atenção a detalhes de modelagem e não enfatiza a importância das dimensões compartilhadas e medidas uniformes.

Os benefícios do modelo Inmon são as seguintes:

- Melhor definição estratégica do projeto;
- Data Warehouse corporativo;
- O processo de ETL é bastante simples;
- Há menor taxa de crescimento de volumes de dados.
- Desenvolve uma abordagem mais vocacionada para definição das componentes de Back End.

#### 4.1.2. O Modelo Kimball

De acordo com Gomes (2011, p. 16), o modelo de Kimball difere em muitos aspetos da aproximação tradicional das Base de dados relacionais. Uma diferença significativa, é que os DWs construídos utilizam a metodologia de modelagem dimensional. Outra diferença importante, é a arquitetura global ser caracterizada por múltiplas Data Marts, que são interligadas entre si pelo Data Warehouse Bus, responsável pela coerência da informação integrante do DW.

As organizações, onde a capacidade de medida previsível, nos lugares mais estáveis, ou seja, onde as dimensões, granularidade e medidas são bem conhecidas e não mudam com frequência, usualmente utilizam este modelo, dado que é detentor de um elevado grau de sucesso. O insucesso do modelo Kimball pode ocorrer se escolher a granularidade errada da primeira vez implementada, e no caso de aparecer uma nova perspectiva de se olhar ou abordar o negócio, que pode custar uma outra implementação e associação de uma nova data mart ao DW.

Kimball apresenta uma metodologia própria para o desenvolvimento de DW. Segundo Albino (2009, p. 20) o design desse modelo assenta num processo que se pode decompor em 4 passos distintos:

### **1. Selecionar o processo de negócio**

Segundo Caldeira (2012, p.155), o processo de negócio “*deve ser encarado como um fluxo de informação que atravessa horizontalmente uma organização, nunca como o resultado de uma análise específica*”. O mesmo autor refere que o processo de um negócio, produz um grupo de medidas, com características específicas de dimensionalidade e granularidade, que deve ser colocado à disposição dos utilizadores, de modo a que este possa analisar os factos, sem ter que recorrer a ajuda especializada. No desenvolvimento do modelo de dados dimensional, a escolha do processo de negócio é considerada como suporte do modelo, e por esta razão, a equipa de implementação do modelo, deve gastar tempo suficiente na identificação do processo de negócio sendo a construção da matriz em bus uma ferramenta poderosa, para prestar auxílio nesta fase. Uma escolha errada do processo de negócio trará grandes consequências, e uma das mais catastróficas, é um modelo de dados nunca terminado e tabelas de factos com grão errado. De acordo com Caldeira (2012, p. 156), um dos melhores métodos de escolha do processo, é um diálogo contínuo com os utilizadores do data warehouse. Antes de escolher o processo de negócio é recomendada a identificação do problema a ser resolvido e a posteriori escolher o processo que engloba o problema a ser solucionado;

### **2. Definir a granularidade, ou atomicidade da informação**

É o processo de escolha do detalhe dos dados contidos no DW. Depois de escolher o processo de negócio, deve ser identificado o nível mais baixo de granularidade, que é designado de grão atómico, significando que não pode ser dividido. Escolher um nível de grão mais atómico possível é o desejável, pois só dessa forma temos a garantia que os utilizadores, podem agregar os dados à sua vontade, e o DW terá a capacidade de resposta maximizada. Se for escolhido um nível intermédio de grão, implica o risco de não ser possível satisfazer todas as pesquisas solicitadas pelos utilizadores do sistema. Nas palavras de Kimball e Ross (2002, p. 34):

*“Preferably you should develop dimensional models for the most atomic information captured by a business process”.*

### **3. Escolher as dimensões.**

Após a escolha do processo de negócio e a definição da granularidade, a próxima fase é escolher as dimensões, onde cada dimensão contém um grande número de atributos. Kimbal defende e aconselha a desnormalização das tabelas dimensões. Caldeira (2012, p. 157) preconiza que a melhor forma de escolher as dimensões que descrevem os factos e as medidas é recorrer à matriz em bus (capítulo 4.7.1). Os atributos nas dimensões são cruciais para

que os factos possam ser facilmente utilizados. “Outras *dimensões podem ser adicionadas ao esquema inicial desde que respeitem a granularidade de tabela de factos. Caso surjam novas dimensões que enriquecem o modelo inicial, não respeitando o grão, então é necessário encontrar um novo grão*” (Caldeira, 2012, p. 157);

#### **4. Identificar as métricas da tabela de factos**

O último passo consiste em determinar as métricas que devem figurar como constituintes nas tabelas de factos. A título de exemplo podemos referir: «Ocupação», «Código da Data Inicial», «Código da Data final», «localidade», são os factos, da tabela de factos do sistema SPARES. Essa tabela tem como “*responsabilidade de armazenar os atributos mensuráveis do processo de negócio que contribuam para a análise do que estamos a medir estabelecer a ligação com as tabelas de dimensão*” (Albino, 2009, p. 20). De registar que, ao contrário da maioria dos sistemas de BI, em que a identificação e determinação dos factos, normalmente são centralizadas em valores numéricos, que preencherão cada uma das linhas da tabela de factos, no caso do sistema SPARES a maioria dos factos é não-numéricos, são dados qualitativos, que segundo Caldeira (2012, p. 158) “*estão sujeitos a múltiplas formas de escrutínio*”.

Kimball salienta os propósitos do data warehouse resumidamente da seguinte maneira:

- Permite aceder às informações com muita facilidade;
- As informações estão organizadas de forma consistente;
- É resistente à mudança;
- Suporte fundamental no processo de decisão

Kimball finaliza os propósitos com uma observação e uma recomendação para qualquer projeto de business intelligence, que é o seguinte:

*"The business community must accept the data warehouse if it is to be deemed successful" (Kimball e Ross, 2002, p. 4).*

No modelo Kimball também há algumas desvantagens, como por exemplo: crescimento vertiginoso de dados na tabela de factos, dificuldade de identificar os factos e granularidade, o processo ETL é muito complexo, mas mesmo assim, traz inúmeras vantagens para o sistema de BI, podendo a título de exemplo mencionar as seguintes:

- A sua estrutura é adequada às exigências de um sistema de apoio à decisão;
- Devido à flexibilidade da estrutura, oferece a facilidade de promover alterações nos sistemas fontes de dados;
- Alto nível de desempenho;
- Dá a garantia de alto envolvimento com o utilizador do sistema;
- Proporciona um melhor retorno sobre o investimento efetuado.

## 4.2. O que é Data Warehouse?

De acordo com Bonel (2015, p. 57), na Data warehouse encontramos todo o histórico da organização ou empresa, uma vez que se ela estiver bem estruturada, as informações são de qualidade. Citando o mesmo autor, este define Data Warehouse como sendo,

*“Um conceito que consiste em agrupar informações por macros assuntos”.*

A DW na visão de Inmon(2002), citado por Gomes (2011, p. 3), é definida como:

*“(...) a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management’s decisions.”*

O conceito de DW baseado na definição de Inmon, realça várias características essenciais a considerar: desde logo é orientado a *Assunto* uma vez que tem como principal objetivo gerar informações sobre um assunto particular, que geram conhecimentos acerca do processo do negócio. Entretanto, para gerar informações confiáveis, é inegável a existência da *Integração* entre os dados, que são reunidos no armazém de dados (DW) a partir de uma variedade de origens e fundidos em um todo coerente e uniforme.

Para além disso, outra característica é a de *Não-volatilidade* da Data Warehouse, no sentido de que os dados são estáveis na DW. Havendo mais dados podem ser adicionados, mas nunca removidos, o que permite uma visão consistente dos negócios. Refere-se justamente ao facto de que nunca devemos efetuar as atualizações no ambiente da DW, sendo aconselhável que as atualizações de dados sejam efetuadas no sistema transacional que dá o suporte ao modelo dimensional.

A última característica a destacar está relacionada com o *Horizonte temporal*, no sentido de que todos os dados contidos na DW são identificados com um período de tempo particular, correspondendo nada mais nada menos, do que a um conjunto de retratos do sistema transacional capturados num determinado período de tempo.

Por fim, a função basilar do DW, consiste na prestação do auxílio aos gestores nos processos decisórios da organização.

Barbosa e Oliveira (2007, p. 29) apresenta o conceito da DW como:

*“O processo de captura dos dados dos sistemas legados e transacionais e a transformação deles em informação organizada, em um formato amigável, para incentivar a análise de dados e para suportar a tomada de decisão baseada em factos do negócio.”*

Em todas as definições vistas, salientamos alguns pontos de vista: o facto do sistema de DW ser separado dos sistemas OLTP, razão pela qual as queries realizadas não alteram a

performance dos sistemas transacionais. A sua implementação, vai depender da natureza do projeto, não havendo nenhuma metodologia predefinida para a sua implementação. A DW é constituída pelos dados oriundos dos vários sistemas de modelos relacionais efetivos na organização. Todavia, é necessário adotar algumas precauções na captura destes dados, já que estes sistemas muitas vezes não estão integrados.

Em suma, e em jeito de síntese, considero que o conceito de DW, traduz de modo inequívoco ser um grande repositório de dados, com um nível alto de integridade, com pouca redundância, que possibilita a análise dos dados, agregando-os de várias formas exequíveis, e que proporciona informações precisas e fundamentais no âmbito de um processo decisório das organizações.

### 4.3. Objetivos e Vantagens da Data Warehouse

Segundo Bonel (2015, p. 56), o funcionamento da DW é considerado uma hierarquia de assuntos, em que o seu objetivo primordial é organizar as informações de forma multidimensional oferecendo em simultâneo a facilidade e optimização das consultas e análise de dados. Caldeira (2012, p. 55) “*refere que, o modelo dimensional surge como uma metodologia que procura atingir dois objetivos principais*”:

- Uma fácil interacção com o utilizador final da aplicação;
- Um alto desempenho no processamento das queries.

Em síntese, pode-se afirmar que este modelo tem um papel de facilitador na interação do utilizador do sistema com as múltiplas dimensões que o compõem, e que estão altamente desnormalizadas.

Este conceito, na ótica de Bonel (2015, p. 57), oferece inúmeras vantagens a um sistema de BI, conforme infra se elenca a título de exemplo:

- **Manter o histórico das informações;**
- **Informações de boa qualidade;**
- **Repositório único para aceder às informações;**
- **Informações agregadas em hierarquias;**
- **Informações consolidadas e otimização nas consultas (rapidez);**
- **Vantagem competitiva**, servindo de auxílio ao gestor, detetando rapidamente as exigências de mercado e identificando os riscos no processo decisório, sendo por estas razões compensador o custo elevado da sua implementação.
- **Simplicidade**, fornece uma imagem simples da realidade da organização;
- **Facilidade de uso**, a maioria das ferramentas de consulta, principalmente o Tableau facilita o acesso aos dados, uma vez que trabalha com interfaces gráficas, o que torna a análise das informações armazenadas no DW uma tarefa intuitiva para os utilizadores.

- **Valores quantitativos**, permite uma retrospectiva realista da evolução da organização, pois possui medidas quantitativas (KPIs), permitindo a comparação e a análise em períodos de vários anos.

#### 4.4. A Arquitetura do Data Warehouse

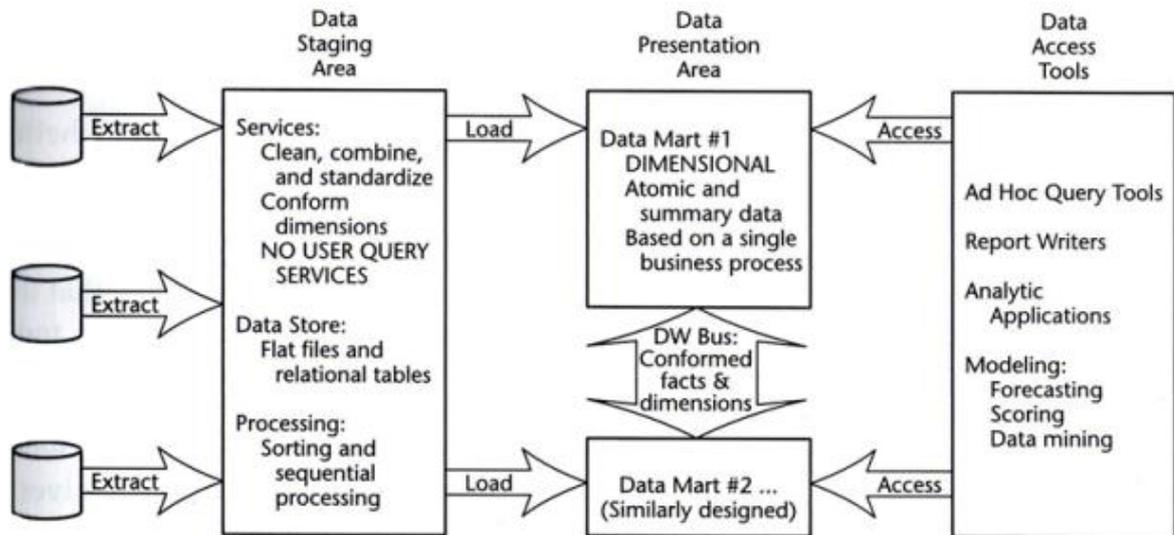


Figura 12 - Elementos básicos da arquitetura da DW do modelo Kimball.

Fonte: (Kimball e Ross, 2002)

A natureza do projeto e a necessidade da organização, são os pontos relevantes na identificação da arquitetura de Data Warehouse, apropriada para satisfazer a necessidade da organização. O sucesso do funcionamento de um sistema de BI, é condicionado pela seleção da arquitetura da DW. Consequentemente, existem vários leques de escolhas da arquitetura, podendo assim optar-se pelo desenvolvimento e implementação de uma DW organizacional transversal a toda organização, ou pela construção de Data Marts específicos para um ramo na organização.

No sistema SPARES a arquitetura do modelo dimensional utilizada, é a exemplificada na Figura 12. É o modelo preconizado por Kimbal, onde a arquitetura de uma DW é formada por um conjunto de DM, que deve traduzir a globalidade da organização de forma coerente.

Outro aspeto importante no processo da implementação da arquitetura da DW, é o grau de cautela que se deve ter na escolha da granularidade (grau de detalhe com que os dados são armazenados num determinado nível), uma vez que quanto maior for o nível da granularidade, menor são os detalhes contidos nos seus dados, e ao invés, quando o grão for o mais atómico possível maior detalhe teremos nos dados.

O Quadro 3 apresenta-nos a tabela de facto como sendo a tabela principal do modelo dimensional, onde são armazenadas as medidas e os factos constituintes do sistema SPARES. Comumente, o termo facto é utilizado para representar uma medida métrica ou quantitativa, mas no caso em concreto, as medidas desse sistema estão intrinsecamente ligadas aos dados

qualitativos. Cada linha da tabela de factos é correspondente à granularidade atómica definida. A tabela de factos pode ser considerada como a alavanca determinante do modelo dimensional.

#### 4.5. Tabela de Facto

Quadro 3 - Tabela de Factos do SPARES, com as novas dimensões.

Comissários	
PK,FK1	<u>Sigla de P2</u>
PK,FK2	<u>Código de P2</u>
PK,FK3	<u>Nome de P2</u>
PK,FK4	<u>Código da Data Inicial</u>
PK,FK5	<u>Código da Data Final</u>
PK,FK6	<u>Ocupação</u>
PK,FK7	<u>Localidade</u>
PK,FK8	<u>Tipo de Evento</u>
	Referência (DD)
	Relação
	Nome de P1
	Sigla de P1
	Código de P1
	Data Inicial
	Data Final
	Evento
	Documento
	Sigla do Projecto
	Notas
	Cotas
	Projeto

Chaves estrangeiras

Factos

As siglas **PK** e **FK** identificam respetivamente, as chaves primárias e as chaves estrangeiras. O atributo “**(Referência)**” DD<sup>15</sup>, é uma dimensão degenerada uma vez que não é métrica nem regista qualquer tipo de facto. Segundo Caldeira (2012, p.121), “A dimensão degenerada funciona, na tabela de factos, como a chave de uma dimensão só que não está ligada a nenhuma tabela. As dimensões degeneradas são usualmente valores numéricos que, em conjunto com as chaves estrangeiras, constituem a chave primária composta da tabela de factos.”

Conforme o Quadro 3 evidencia, a tabela de factos «Comissários», possui oito chaves estrangeiras, surgindo reportadas a cinco dimensões, já que existem chaves estrangeiras que pertencem à mesma dimensão. As tabelas satisfazem a sua integridade relacional, dado que as chaves na tabela de facto são compatíveis com as suas respetivas chaves primárias nas tabelas dimensão correspondentes. Por exemplo, o «Código da Data Inicial» na tabela de facto é sempre compatível com a chave Código da data na dimensão Data.

<sup>15</sup> DD – Dimensão Degenerada

Dentro da tabela de facto, temos diferentes tipos de factos que são: Aditivos, Semi-Aditivos e Não Aditivos. Os factos aditivos são os que podem ser somados para todas as dimensões interrelacionadas com a tabela de facto, como por exemplo o número de produtos vendidos num talão de recibo. Os Semi-aditivos são factos que apenas são aditivos face a algumas das dimensões que participam na tabela de factos e não à sua totalidade. Os factos não aditivos – são as medidas normalmente sem características aritméticas, e que não se podem somar de todo (Albino, 2009, p. 24).

Normalmente, os factos numéricos e aditivos têm uma maior usabilidade. Sabendo que a agregação, de acordo com Caldeira (2012, p. 69), “... é o mecanismo de cálculo que a partir de um facto com um alto grau de atomicidade produz um resultado sumariado, esta operação é uma forma de incrementar a velocidade de exploração dos dados em data marts dimensionais.”, o que faz com que seja um processo aditivo, e uma das operações com maior usabilidade num Data Mart.

A agregação dos dados permite facilitar a sua consulta e a obtenção de resultados e uma visão dos mesmos sob diferentes perspetivas ou dimensões. A título meramente exemplificativo podemos citar a análise de ocupações, cujos dados podem ser agregados através de tipos de setores, tipos de estatutos, ou tipos de ocupações, efetuando em consequência a exploração do Data Mart em menor tempo possível. Em termos de performance, as tabelas de factos são altamente normalizadas, com pouca redundância dos dados, o que as torna bastante eficazes.

#### 4.5.1. Tipo de Tabelas de Factos

Existem três tipos distintos de tabelas de factos:

- As **Tabelas Transacionais** que, de acordo com Caldeira (2012, p.72), “registam os factos que se vão medindo em múltiplos pontos discretos ao longo de um determinado período de tempo. Esta categoria de tabelas é a mais vulgar no modelo dimensional pois é o reservatório central de cada data mart”. Como o próprio nome indica, o nível de granularidade é correspondente a cada transação. Esse tipo de tabela, permite dar resposta a todas as análises de dados efetuadas, uma vez que, o seu grão é o mais atómico possível. É esse tipo de tabela que é utilizado no sistema SPARES;
- **Tabelas de Sumarização Periódica** “agregam as métricas de um processo de negócio em intervalos de tempo regulares e pré-estabelecidos. (Caldeira, 2012, p. 72)”. Resumidamente, esse tipo de tabelas armazenam um registo para cada período de tempo pré-determinado já com a agregação dos dados, ou seja, os registos são sumarizados de acordo com um determinado intervalo de tempo estabelecido, de vital importância para a organização. Normalmente, se existir uma dimensão temporal de data ou hora a sua granularidade não será o mais atómico possível, mas sim o período de tempo acordado;
- **Tabelas de Sumarização Acumulativa** “são de facto as menos utilizadas no data warehousing. Este tipo de tabelas é independente do tempo e cobre o ciclo de vida completo de um acontecimento contínuo ou discreto.” (Caldeira,

2012, p. 72). Esse tipo de tabela não é tão comum com os outros dois, sendo que o intervalo de tempo para preencher um registo não é determinado, e apresenta em cada registo todas as etapas de duração de uma determinada atividade.

Para obter um sistema de data warehousing mais compacto, completo e robusto, é uma mais-valia conseguir associar os três tipos de tabelas de facto supra referenciados, no data marts, devido ao facto de fornecer uma visão de todos os ângulos sobre o processo do negócio.

#### 4.6. Dimensões

As dimensões são tabelas participantes no processo de negócio em estudo. As tabelas de dimensão são diferentes do modelo relacional, porque são consideradas tabelas com uma chave artificial, que é um campo que serve de chave primária numa dimensão, e constituídas por muitos atributos. Segundo Caldeira (2012, p. 92) *“As tabelas de dimensão são, de uma forma geral e de acordo com a teoria da normalização, altamente desnormalizadas. Por outro lado, ocupam mais espaço no data warehouse, normalmente entre 10 a 15 % do total dos dados armazenados”*.

Quadro 4 - Dimensão Localidade do SPARES

Localidade	
PK	Referência
	Localidade
	Província
	Bispado
	Concelho
	Sede de Concelho
	Comarca
	Freguesia
	Termo
	Latitude
	Longitude
	Ano
	Fonte
	Notas
	Capitania
	Ilha
	Lugar
	Arquipelago

As tabelas de dimensão complementam a data warehousing, sendo membros essenciais de uma tabela de facto. Habitualmente contêm as descrições textuais do processo

de negócio, numa forma muito bem detalhada, que caracterizam as medidas existentes na tabela de factos.

*“Tipicamente, os atributos de uma dimensão são descritivos e não-mensuráveis, traduzindo-se em valores textuais com um número limitado de possibilidades. É também habitual que estes atributos possam ser organizados segundo hierarquias bem definidas, o que possibilita a análise dos factos a diferentes níveis de granularidade. Os valores possíveis para cada nível da hierarquia denominam-se membros”* (Antunes, 2008, p. 11).

O Quadro 4 retrata, como exemplo, uma dimensão «localidade», e seus atributos descritores. Nota-se que a Surrogate Keys «Referência» não tem nenhum significado específico para a localidade, mas essa chave traz inúmeras vantagens, como por exemplo, a facilidade de integração de fontes de dados heterogêneas; a introdução de registos que não existam no sistema transaccional; a sua independência face às chaves do sistema transaccional; na alteração de um registo garante a unicidade da chave; e na performance, por utilizarmos um número inteiro para a chave primária, é sempre maximizada, e o espaço em disco minimizado. (Albino, 2009, p. 28)

Os outros atributos da dimensão localidade apresentados no Quadro 4, têm como finalidade auxiliar na descrição da localidade, tendo como exemplo: a província que pertence a tal localidade, o seu concelho, a sua georreferenciação «latitude» e «Longitude», entre outros. Todos esses atributos descritores não são de preenchimento obrigatório, uma vez que podem ser desconhecidas algumas das descrições da localidade. Consequentemente, as dimensões são geometricamente menores que as tabelas de facto.

Citando Caldeira (2012, p. 92), uma dimensão é considerada conforme quando obedece a duas regras básicas na sua implementação:

- Abrangência para mais do que um processo de negócio;
- Tem que possuir um conjunto completo de atributos, como tomar a devida precaução com as denominações dos atributos, principalmente escrever os nomes completos, num português correto, e com o respetivo significado. Por exemplo, o atributo «Latitude» não deverá ter outra denominação, pois tal carecerá de sentido.

#### **4.6.1. Dimensão Lixo**

Para além dos elementos referidos existe ainda uma outra classe de dimensões, a dimensão lixo<sup>16</sup>. Uma dimensão deste tipo, é apenas uma forma conveniente de agrupar os atributos associados aos factos, mas que não são características de nenhuma das dimensões identificadas, e que não traduzem entidades por si só, ou seja, uma dimensão lixo não é mais do que uma coleção de códigos aleatórios transaccionais, atributos de texto que são alheios a qualquer dimensão. A dimensão lixo constitui assim uma estrutura, que fornece um local conveniente para armazenar os atributos chamados de ‘lixo’ em uma modelagem dimensional.

<sup>16</sup> Junk Dimension, em inglês.

A criação de uma dimensão deste tipo evita a proliferação de dimensões degeneradas, reduzindo o espaço ocupado pela tabela de factos e evita igualmente a perda de informação.

Segundo Caldeira (2012, p. 145), as principais características de uma dimensão lixo é a sua descontextualização, devido ao facto de ter uma baixa cardinalidade e de não encaixar em nenhuma dimensão do sistema. Os atributos dessa dimensão em algumas ocasiões são importantes para ter uma informação adicional relevante para a análise de alguma área no negócio.

A título de exemplo prático podemos referir num sistema de vendas, a dimensão «forma de pagamento» (multibanco, dinheiro vivo, etc...) que é considerada útil, porque fornece informação adicional sobre um acontecimento. A razão dessa dimensão ser considerada como dimensão lixo, está ligada ao facto de para o gestor não ser relevante saber a forma do pagamento para a tomada de qualquer decisão, mas é revelante para a tabela de factos, diminuindo o acréscimo exponencial dos dados.

Esse constitui um dos exemplos mais simples de exemplificar e auxiliar o utilizador a perceber a importância dessa dimensão. A conceção de uma dimensão desse género foi um dos casos de estudo - **será possível implementar uma dimensão lixo no sistema SPARES, auxiliando a tabela de factos a não ter um crescimento exponencial dos dados?**

Chegou-se à conclusão de que neste momento não será possível criar a dimensão lixo, devido à natureza do sistema SPARES, onde algumas dimensões mesmo não sendo uma dimensão lixo, ocupam em parte o papel dessa dimensão.

#### 4.6.2. Dimensões Role-Playing

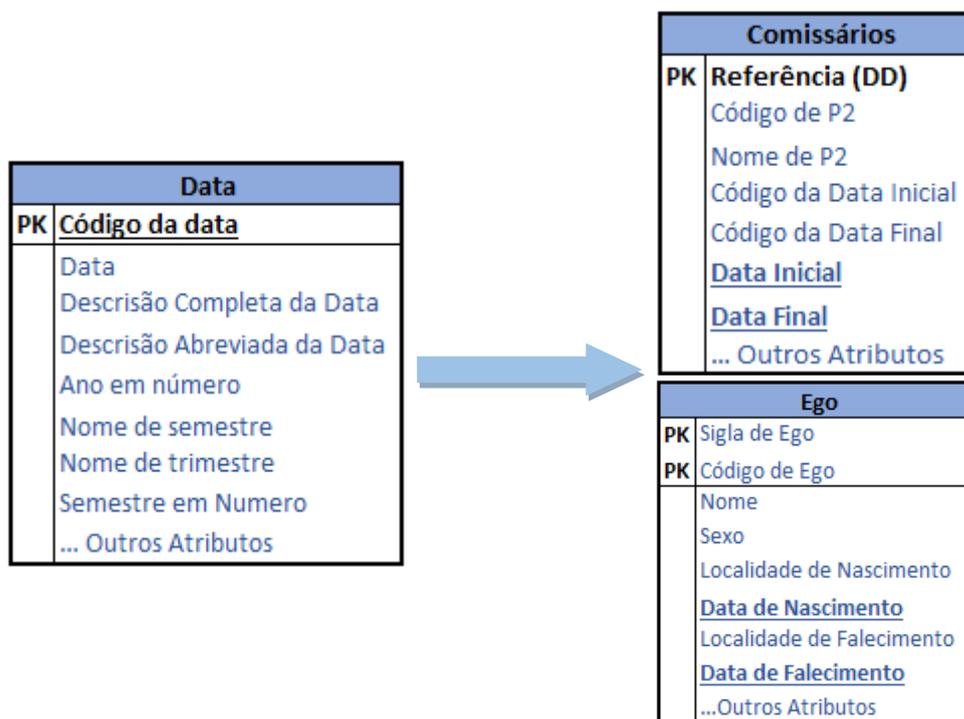


Figura 13 - Exemplo de dimensão Role Playing

As dimensões *role-playing* exercem diversas funções dentro do data warehouse, consoante o contexto em que sejam aplicadas. Um bom exemplo de uma dimensão *role-playing* é a dimensão data conforme a Figura 13 apresenta, devido ao facto da frequência da existência de datas com significados diferentes.

No sistema SPARES, a dimensão «arquivo» tem a data inicial, data final do arquivo, e data da última alteração do arquivo, enquanto que na tabela de factos temos a data inicial e a data final de um dado evento registado. Na dimensão «documento» também há, a data inicial e a data final de um documento registado, enquanto que a dimensão ego apresenta outras perspetivas da data, como a data de nascimento e data de falecimento, complementado com a data de casamento na dimensão de genealogia.

Outro exemplo é a dimensão «localidade». Na tabela de factos há um atributo que refere a localidade onde aconteceu o evento, mas a localidade pode ter outra perspetiva na dimensão de ego, registando a localidade de nascimento e localidade de falecimento de uma determinada pessoa.

#### 4.7. Data Mart Ocupações

Data Mart é uma base de dados peculiar, aplicável a uma determinada área específica dentro da organização. Pode ser uma base de dados acerca de um departamento, ou assunto particular, salientando-se a simplicidade de aceder a informação pertinente. Por outras palavras, Data Mart é um subconjunto de um Data Warehouse, dispondo das mesmas propriedades.

Caldeira (2012, p. 204), define Data Mart como, “*um conjunto de dados desenvolvidos com suporte num esquema em estrela e que pertencem à área de publicação do data warehouse. No passado, os Data Marts eram concebidos como uma forma de apresentar dados unicamente agregados, i.e., sobre os quais era aplicada uma aritmética qualquer, mas isso produzia aplicações muito rígidas que conseguiam responder apenas a um conjunto muito limitado de questões.*”

O mesmo autor defende, que atualmente os Data Marts constituem uma estrutura muito flexível, onde há uma preferência evidente por se incorporar os dados o mais atómicamente possível. As suas características são idênticas a uma data warehouse organizacional, sendo os dados são apresentados aos utilizadores na forma de esquema em estrela.

De acordo com Sezões, Oliveira, e Baptista (2006, p. 39), a criação de um data mart tem inerente como imperativo o seguinte:

- **Performance.** Consiste em separar um conjunto de dados do DW corporativo com vista a uma maior eficiência no seu tratamento e processamento de dados;
- **Segurança.** Nomeadamente é separar as informações autorizadas para determinado grupo de utilizadores de informação confidencial;
- **Utilidade.** Necessidade de um modelo de dados diferente do DW para uma finalidade de negócio diferente.

Resumidamente, Data Mart é um pequeno DW, ou seja, um armazenamento de dados em escala menor do que DW, com finalidade de fornecer suporte à decisão numa organização. Os Data Marts atendem às necessidades de áreas específicas de negócios, ao invés da corporação como um todo. Tem vários benefícios, sendo de destacar essencialmente a diminuição substancial do custo de implementação e manutenção de sistemas de apoio às decisões.

#### 4.7.1. Matriz em Bus

De acordo com Caldeira (2012, p. 175), “A matriz em bus não é uma peça decorativa. É um elemento essencial na construção de qualquer data warehouse, e se a sua construção e utilização forem desprezadas, as organizações acabam por ter não só um repositório de dados devidamente formatado para análise da informação, mas um conjunto de tulhas com dados díspares e sem nenhuma utilidade empresarial.”

Processos	Tipo	Dimensão					Dimensão
		Data	Ocupação	Localidade	Tipo de Sector	Tipo de Estatuto	
<b>Agrupamento de ocupações por sectores de atividades, tipos de ocupações e tipos de estatutos sociais, num determinado Período tempo, e numa determinada Localidade</b>							
Análise temporal (século XVI), qual a ocupação agrupado por tipo de estatuto que predominava, numa localidade	Sumarização Periódica	X	X	X		X	4
Análise temporal (século XVII), a ocupação agrupado por tipo de ocupação	Sumarização Periódica	X	X			X	3
Análise temporal (século XVI), qual a ocupação agrupado por tipo de sector que predominava, numa localidade	Sumarização Periódica	X	X	X	X		4
Análise de ocupação por tipo de sector, tipo de ocupação e tipo de estatuto numa localidade	Transaccional		X	X	X	X	5

Figura 14 - Matriz em Bus sobre o processo de análise de ocupações

Segundo Caldeira (2012, p. 172), “matriz em bus é a ferramenta utilizada para criar, documentar e comunicar a arquitetura do data warehouse”. Cada linha da matriz representa um processo. Como o seu próprio nome indica, é uma matriz ou um quadro, que tem como seu principal objetivo, prestar auxílio como base ao desenvolvimento consistente de cada fase do data mart (Figura 14).

O mesmo autor refere que, a matriz em bus é uma arquitetura essencial para a construção, desenvolvimento e implementação do data warehouse. As linhas da matriz são definidas após a delimitação de todos os temas analíticos do processo de negócio, onde cada uma dessas linhas vai corresponder uma área fundamental do funcionamento da organização.

A dimensionalidade de cada processo vai depender das colunas associadas, ou seja, as dimensões ligadas a essa área de pesquisa. Dessa forma, permite que o processo essencial do

sistema SPARES tenha maior dimensionalidade que os outros processos, dado que tem quase todas as dimensões associadas.

Infra, cita-se a recomendação de Caldeira (2012, p. 174):

*“Nunca se deve estrangular o desenvolvimento do data warehouse optando inicialmente por um pequeno número de tabelas de dimensão. Uma lista extensa de dimensões é sempre o melhor ponto de partida para o correto preenchimento da matriz”.*

Dado a estrutura simplificada da matriz, a matriz em bus é uma ferramenta de vital importância para comunicar com os utilizadores do sistema, a fim de se perceber o funcionamento dos processos de negócio. Na fase de desenvolvimento do data warehouse, serve de base para a implementação de cada data mart constituinte do DW.

Se repararmos, a arquitetura em bus não obriga a que todas as dimensões sejam analisadas pelo mesmo conjunto de perspetivas, mas sim, que todas as perspetivas de análise de um processo são únicas. Por exemplo, na DW existirá uma única dimensão temporal «data» (tipicamente com a granularidade mínima diária), e uma só dimensão geográfica «localidade» (tipicamente com a granularidade mínima correspondente a um ponto geográfico), em que cada uma delas pode estar associada a um processo de negócio ou não.

Em suma, as linhas da matriz correspondem a data marts e as colunas a dimensões conformadas. A matriz é a ferramenta usada para criar, gerir e comunicar a arquitetura do data warehouse. Segundo Kimball, é o artefacto de análise mais importante do desenvolvimento de um DW. É uma ferramenta híbrida, que serve para design técnico do DW, e uma forma de comunicação organizacional.

#### **4.7.2. Implementação de Data Mart Ocupações**

Há duas formas através das quais podemos implementar os data marts. Em primeiro lugar, podemos criar um data warehouse centralizado que contém todos os dados juntos, provenientes de várias fontes de dados da empresa, implementando a posteriori múltiplas data marts dependentes como subconjuntos desse data warehouse central.

A outra forma de implementação, é criar um data mart independente para cada processo de negócio e, no caso de haver a necessidade de efetuar a integração de todos os data marts, e desta forma termos um data warehouse mais abrangente. O data mart ocupação foi utilizado na primeira forma para seu desenvolvimento e implementação.

Os data marts dependentes têm a vantagem de se conseguir criar um modelo de dados com maior consistência, simplificando a garantia absoluta da qualidade de dados, uma vez que os dados são todos extraídos, transformados e carregados pelo mesmo processo de ETL, e são organizados na mesma data warehouse.

No entanto, a construção de um data warehouse centralizado e completo antes da implementação dos data marts e conseqüentemente das aplicações e relatórios, não é a melhor

solução em empresas de grandes dimensões que estão em constante mudança e a obter novas fontes de dados (Gouveia, 2013, p. 14).

		Setor de atividade																					
		Administração	Agricultura	Belas Artes	Defesa	Diplomacia	Doméstico	Eclesiástico	Ensino	Justiça	Mineração	Não se sabe	Navegação	Negócio	Pecuária	Pesca	Saúde	Segurança	Sem ocupação	Sem ocupação/nobre	Transformação	Transporte	Viver da sua fazenda
Ocupação (designação da época)	Artes liberais	●						●	●								●						
	Artes mecânicas							●									●				●	●	
	Belas Artes			●																			
	Embaixadas					●																	
	Estudo							●															
	Governo da casa	●					●					●											
	Governo da res publica	●			●			●		●								●	●				●
	Igreja							●															
	Lavoura		●													●							
	Mercancia													●									
	Milícia				●														●				
	Mineração											●											
	Não se sabe	●											●										
	Navegação													●									
	Pesca															●							
	Sem ocupação																		●				
	Sem ocupação/nobre																			●			
Viver da sua fazenda																						●	

Figura 15 - Agrupamento de ocupações (designação da época) por setores de atividade.

Fonte: (Ferreira, 2016, p. 26)

As ocupações encontram-se associadas a diferentes estatutos sociais. Apresenta-se na Figura 15 “os tipos de estatuto possíveis para cada ocupação/setor de atividade. O estatuto associado a um determinado indivíduo pode ser mecânico, não mecânico, nobre ou ser desconhecido. O estatuto é Mecânico ou Não Mecânico consoante o trabalho dos indivíduos seja, ou não, manual. Mecânico corresponde ao oposto de nobre. Remete para aquele que trabalha com as mãos e que tem de trabalhar para sobreviver. Deste modo está “mais abaixo” na hierarquia social um indivíduo cujo estatuto é mecânico. Não mecânico é aquele que não é nobre, mas também não é exatamente mecânico ou quando não se tem a certeza que é nobre. O estatuto Nobre corresponde a todos os indivíduos que tenham um título de nobreza. Sempre que o estatuto é desconhecido, considerou-se que o estatuto era “Não Se Sabe”. (Ferreira, 2016, p. 26).

A Figura 15 serviu de fundamento para o processo ETL da data mart ocupações. Foi criado um ficheiro excell com todas as ocupações registadas na tabela de factos, com as

respetivas hierarquias, de tipos de setores, tipos de estatutos e tipo de ocupações de acordo com a época em atuavam os comissários de Santo Ofício.

#### 4.7.3. Processo ETL da Data Mart Ocupações

De acordo com Kimball (2004, citado por Barbosa e Oliveira (2007, p. 31)), existem algumas regras que deverão ser seguidas no processo ETL, de modo a que o Data Mart, ou a Data Warehouse obtenha sucesso. A área de estágio deve ser conhecida apenas pelos administradores do processo ETL. Não são permitidos acessos de utilizadores à área de preparação, devendo essa área ser considerada como um local de construção. O acesso por utilizadores não autorizados, poderá causar danos à qualidade dos dados, podendo afetar de modo irreversível a integridade dos dados na DW.

O processo ETL neste projeto, permitiu a nível pessoal, a comprovação de que este procedimento ocupa mais de 80 % do trabalho no desenvolvimento de um sistema de BI. Tendo surgido a necessidade de implementação de uma Data Mart sobre ocupações, uma vez que é um elemento de grande importância para os investigadores e os utilizadores do sistema SPARES, o processo ETL para esse Data Mart, revestiu-se de alguma complexidade acrescida, pelo facto dos dados deverem ser retirados na tabela de factos «*Comissario*», dentro do atributo «*Evento*», seleccionando-se os que contêm a etiqueta **Ocupação**, como por exemplo:

- `<div>&lt;Ocupação&gt;Bispo do Funchal&lt;/Ocupação&gt;</div>`
- `<Ocupação><div>Lavrador </div></Ocupação>`

Em primeiro lugar, criou-se uma tabela auxiliar contendo todos os registos da tabela de facto, cujo atributo evento contém a etiqueta de ocupação. Em seguida, foi elaborado um procedimento que efetua um ciclo consoante o número de registos da tabela (Aproximadamente 25000), que lê cada registo e em seguida efetua a limpeza desse dado, que consiste designadamente em remover todas as palavras reservadas, `<Ocupação><div>`, `<div>&lt;Ocupação&gt;` tendo surgido, entre outros, um novo atributo com as ocupações da seguinte forma:

- Bispo do Funchal
- Lavrador

A partir dessa tabela auxiliar, através de um *select distinctrow*, obtemos uma nova tabela com cerca de 7000 registos. Esses dados, são as ocupações de carácter distintivo, por exemplo «Bispo do Funchal», «Bispo de Évora», «Lavrador», «Lavrador na freguesia de Santo António - Madeira», etc. Exportamos os dados para uma folha de Excel, e efetuou-se a limpeza dos dados, do seguinte modo: «Bispo do Funchal», «Bispo de Évora», as suas ocupações é igual «Bispo», tendo o tipo dessa ocupação, o estatuto social e o sector de atividade, sido preenchidos de acordo com a época moderna. As palavras-chave obtidas manualmente, de acordo com a tese de Ferreira (2016, p. 73-94), constituem a base informacional para preencher os requisitos referente a ocupação.

Com os dados do ficheiro Excel, exportamos como arquivo csv, para uma nova dimensão denominada de «*Ocupação*». Essa dimensão vai estar ligada com a tabela de facto, oferecendo uma nova face ao cubo multidimensional, e proporcionando pesquisas sobre ocupações que antes não era possível determinar, dado que as pesquisas eram realizadas com filtros avulsos. Com essa nova dimensão, vamos poder analisar os dados podendo parametrizar ou agrupar, através do tipo de ocupações da época, o tipo de estatuto social, e também através do tipo de setor.

Em suma, esse processo ETL para DM Ocupações, foi tipicamente composto pela sequência de onze tarefas recomendadas por Antunes (2008, p. 39):

1. Extração dos dados das dimensões e registo dos metadados respetivos;
2. Extração dos dados referentes aos factos e registo dos metadados respetivos;
3. Processamento das dimensões;
4. Processamento dos factos:
  - 4.1. Verificação da integridade referencial, que é a restrição estabelecida entre duas tabelas com o objetivo de manter a consistência dos dados entre as linhas dessas tabelas;
  - 4.2. Processamento de registos falhados;
5. Carregamento das dimensões;
6. Carregamento dos factos atómicos;
7. Carregamento dos factos agregados;
8. Revisão do processo de carregamento;
9. Atualização dos índices e instante do carregamento;
10. Registo dos metadados referentes ao controlo de tarefas;
11. Verificação do sucesso do carregamento dos dados.

#### 4.7.1. Hierarquias

De uma forma geral as hierarquias são relações que se podem instituir entre os atributos de uma dimensão, permitindo a agrupação das informações, de modo a conseguirmos realizar as análises partindo de um atributo que consideremos como Pai, para outro atributo considerado Filho, sumarizando a informação de uma forma automática.

Essas relações entre atributos, Albino (2009, p. 29) cita Cortes (2005) parafraseando que elas ocorrem devido à desnormalização das tabelas dimensões, ao conter informações na mesma tabela que não se relacione apenas com a chave primária da mesma, leva a que seja possível criar essas relações hierárquicas entre os atributos.

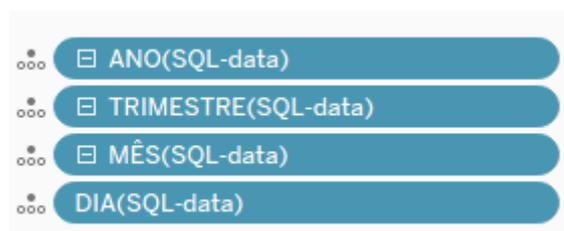


Figura 16 -Estrutura hierárquica na dimensão Data

Observando a Figura 16, verificamos a possibilidade de consultar a data pelo «ano» e efetuar uma operação de roll-up para visualizar em «trimestre», e assim por diante em «mês», até chegar ao nível mais específico, a sua granularidade «dia».

As operações disponíveis pelas ferramentas OLAP para serem utilizadas com as hierarquias, são nomeadamente, o Roll-Up e o Drill-Down. O Roll-Up é a operação através da qual o utilizador efetua a agregação dos dados a observar num nível de granularidade mais saliente ou elevado, e transpor do nível em análise para o seu antecessor hierárquico. Na Figura 16 podemos dar o exemplo de passagem de análise dos dados do nível do dia para análise ao nível do mês. A operação de Drill-Down oferece ao utilizador um pormenor maior do que aquele com que inicialmente observava a série, passando assim do nível em análise para o nível dos seus descendentes. A título de exemplo, a passagem de análise dos dados do nível do trimestre para o nível mês (Ferreira, 2012, p. 12).

Os utilizadores podem precisar de listas de itens que contenham uma estrutura hierárquica. Por exemplo, a dimensão localidade pode conter os atributos Continente, País, Cidade, Freguesia, Localidade. Além da dimensão data e localidade, a dimensão ocupação também contém níveis hierárquicos, podendo, portanto, uma ocupação ser agrupada por tipos de setores, tipos de estatutos ou tipos de ocupações, e cada um desses tipos contém uma lista de filhos. A título de exemplo, podemos citar que o nível hierárquico de tipo de estatuto contém 4 filhos (mecânico, não mecânico, nobre e não se sabe).

As novas dimensões no sistema SPARES, nomeadamente a dimensão «data» e «ocupação», são constituídas por atributos que se distribuem em vários níveis hierárquicos. Podemos definir a hierarquia como sendo um conjunto de associações entre os dados.

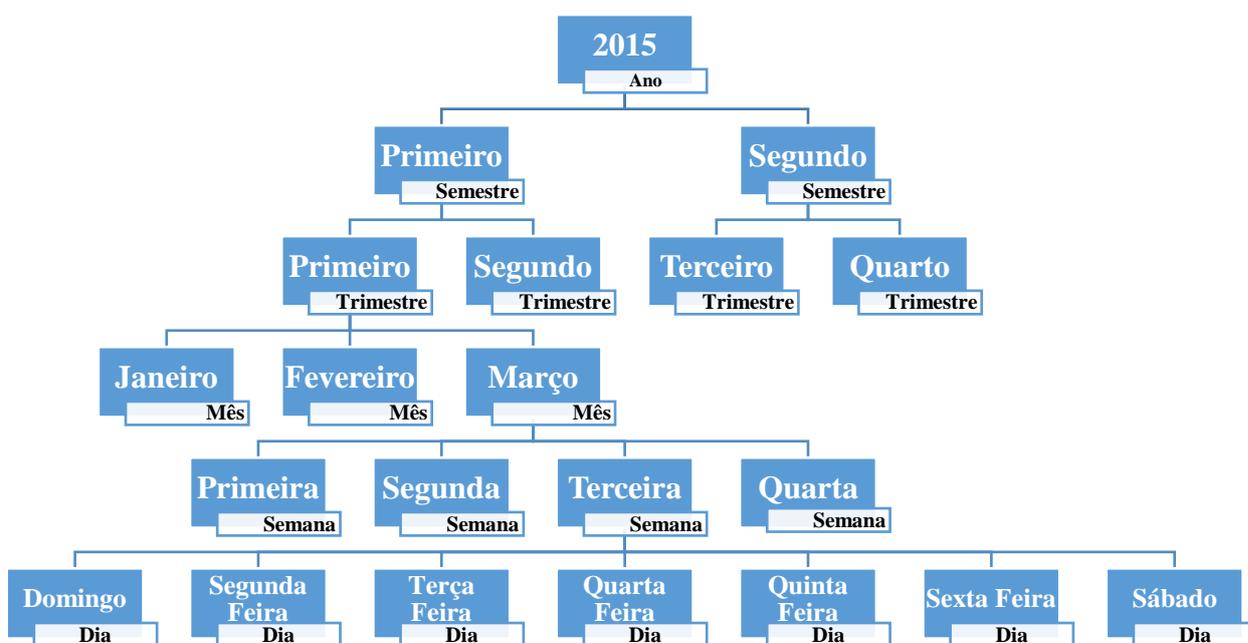


Figura 17 - Conceito de hierarquia na dimensão data

As hierarquias descrevem a lógica dos relacionamentos entre os dados, constituindo a base para a navegação entre os diferentes níveis de detalhe em uma estrutura multidimensional. Caldeira (2012, p. 69) refere que numa *“hierarquia cada atributo é pai ou filho de um outro atributo, e esta relação pai-filho proporciona diferentes graus de sumarização dos dados”*.

A forma de restrição ou associação de dados no modelo dimensional denomina-se de Hierarquia. O exemplo da Figura 17 contém uma hierarquia baseada no horizonte temporal, nomeadamente a data. Deste modo, os dados podem ser sumarizados periodicamente em factos anuais, semestrais, trimestrais, mensais, semanais e diárias. A cada um destes atributos corresponde um nível na hierarquia, concluindo-se assim que temos 5 níveis de hierarquias na dimensão data.

Outros exemplos de hierarquias podem ser os organogramas organizacionais, diferentes níveis geográficos, as classificações taxionómicas, o conjunto de atributos de morada, entre outros.

Caldeira (2012, p. 103) alude a que, *“as hierarquias são uma forma de agrupar dados que representam diferentes categorias numa única dimensão. Elas são os pontos fulcrais para sumarização dos dados de acordo com diferentes perspetivas e grau de pormenor”*.

Geralmente as hierarquias são representadas através de árvores de nós, com vários níveis que descendem de um nó superior designado usualmente por nó pai. Cada nível da árvore é designado por nível dimensional e representa um determinado nível de detalhe na cadeia de agregação associada com a hierarquia. A sequência de todos estes níveis é referida como caminhos de agregação, e o seu tamanho é determinado pela cardinalidade de níveis de agregação que contém.

Visto que, a cada nível corresponde uma coluna, então as hierarquias exigem uma representação simplificada. Caldeira (2012, p. 104) prenuncia que *“a análise de estruturas hierárquicas, nomeadamente as mais complexas, é uma das tarefas mais difíceis na modelação dimensional”*. O autor recomenda que a cardinalidade máxima de níveis hierárquicos numa dimensão não pode ultrapassar cinco níveis.

Segundo Ferreira (2012 p. 10) existem várias metologias para definir as hierarquias num modelo dimensional, entre as quais podemos referir as seguintes:

- Os utilizadores e o desenvolvedor do modelo dimensional, definem em conjunto cada nível de hierarquia, a sua ordem e os seus respetivos atributos de acordo com a natureza da exploração dos dados. Este método, normalmente, conduz a processos evolutivos, pois podem ser afinados com a utilização, experiência e novas necessidades. A intervenção de utilizadores com conhecimento da área é fundamental neste processo. Este foi o processo utilizado no sistema SPARES.
- Os utilizadores especificam unicamente o que consta em cada nível da hierarquia, deixando a ordem dos níveis a cargo das ferramentas OLAP. Os motores de ferramentas OLAP apresentam uma proposta de ordem dos níveis dentro das hierarquias, com base no número de elementos de cada nível. Em

casos especiais, como a dimensão tempo, esta poderá ser definida por uma estrutura pré-concebida.

- Os motores de ferramentas OLAP têm a capacidade de identificar os elementos que definem cada nível da hierarquia e a ordem dos diferentes níveis da hierarquia. Além de definir automaticamente a ordem dos níveis, definem ainda os elementos e os seus antecessores, com base nos dados fornecidos.

Observando a Figura 17, constatamos que um nó pode ser simultaneamente um nó pai e um nó filho dependendo do nível da árvore em que está estabelecido. O primeiro nó pai representa a vista mais generalizada dos dados. Em suma, a hierarquia é uma estrutura em árvore que organiza os membros de uma dimensão, de modo a que cada membro tenha um membro ascendente e zero ou mais membros descendentes.

### 4.8. O Esquema em Estrela

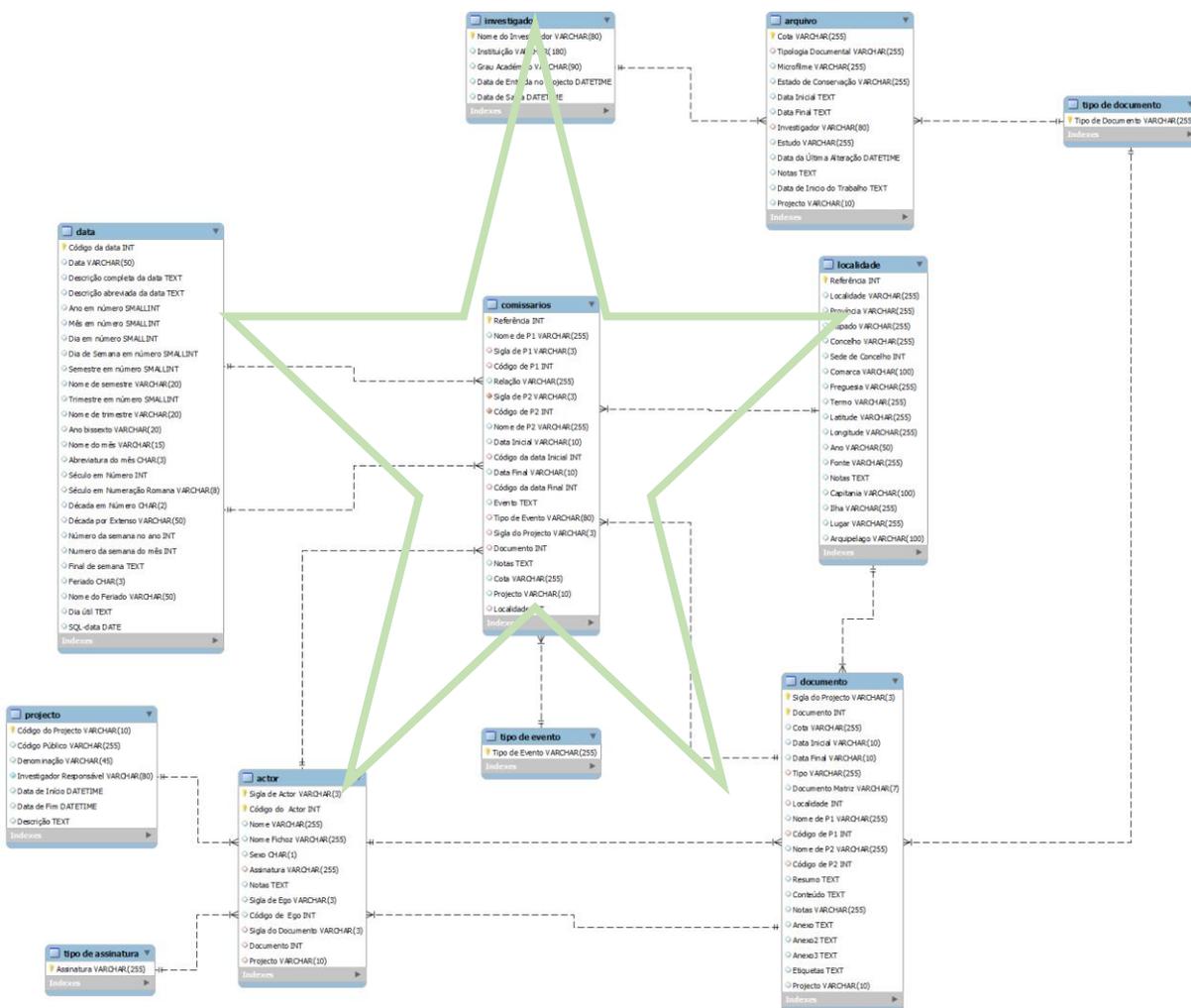


Figura 18 - Esquema em Estrela do SPARES

Após a identificação dos componentes essenciais do modelo dimensional (Tabela de Factos e Dimensões), faltará associar esses elementos de modo a dar origem ao modelo. A descrição de um modelo dimensional é realizada por intermédio de um esquema, que estabelece as relações existentes entre a tabela de facto e as dimensões. O esquema em estrela é utilizado para representar o modelo multidimensional.

O modelo dimensional do sistema SPARES está representado sob a forma de um esquema em estrela e, como se pode observar Figura 18 apresenta uma grande vantagem em relação ao modelo relacional, especificamente nas pesquisas onde são envolvidas múltiplas tabelas, pois permite um maior desempenho e velocidade na análise e exploração dos dados.

O esquema em estrela<sup>17</sup> é o esquema mais simples, e consiste apenas numa tabela de factos ligada a um conjunto de tabelas de dimensão. Neste tipo de esquema todas as tabelas de dimensão estão unicamente relacionadas com tabelas de factos, não existindo qualquer relação entre as tabelas de dimensão.

A utilização de um Esquema Estrela para processamento analítico possibilita alguns benefícios em relação a uma estrutura relacional, de que se destacam os seguintes aspetos:

- O projeto de base de dados multidimensional prevê um rápido tempo de resposta;
- Permite otimizar o sistema para trabalhar com um projeto mais simples de base de dados melhorando a execução do planeamento;
- Permite projetar a base de dados, do modo como o utilizador final habitualmente pensa e usa os dados analiticamente;
- Simplifica o entendimento e a navegação dos metadados por desenvolvedores e utilizadores finais;
- Possibilita um maior número de ferramentas de acesso aos dados, sendo que alguns produtos disponíveis no mercado exigem o projeto de um Esquema Estrela.

O conceito de esquema em estrela pressupõe a criação de tabelas dimensionais. A título exemplificativo temos a dimensão ocupação, dimensão data, dimensão localidade, que ficam ligadas entre si através de uma tabela de factos (comissários). A sua interligação baseia-se num esquema lógico e simples: as tabelas dimensionais contêm as definições das características dos eventos, enquanto as tabelas de factos, por sua vez, armazenam os factos decorridos e as chaves estrangeiras para as características respetivas que se encontram nas tabelas dimensionais.

*“Este modelo apresenta vantagens óbvias, como por exemplo a existência de uma única tabela de factos contendo toda a informação sem redundâncias, a definição de apenas uma chave primária por dimensão, a redução do número de interligações e a consequente pouca necessidade de manutenção”* (Sezões, Oliveira, e Baptista, 2006, p. 36).

---

<sup>17</sup> **Star Schema** em Inglês.

Segundo Ornai (2014, p. 16) o benefício do modelo em estrela, supera quaisquer deficiências, pelo facto do número de operações de “joins<sup>18</sup>” entre tabelas ser minimizado na generalidade das análises efetuadas. Ou seja, este modelo de estrela (Star Schema) é mais simples e oferece uma maior facilidade na navegação dos dados.

#### 4.9. Conceito e Importância dos Metadados

Costa (2012, p. 33) cita (Inmon, 2005) dizendo que, “*associados aos sistemas de data warehousing, encontram-se também os metadados*”. O significado literal de metadados (Metadata) é «dados sobre os dados», isto é, o termo refere-se à informação que descreve as características de um conjunto de dados.

Normalmente, os metadados apresentam-se como: (I) metadados de negócio e (II) metadados técnicos. Os primeiros têm valor para os gestores, os segundos apresentam valor para os técnicos que desenvolvem o sistema.

*“Os metadados, ou informação sobre os dados, são um dos aspetos fundamentais no data warehouse, dada a sua importância para a construção, manutenção e utilização do repositório de dados. Como o data warehouse pode ser derivado de múltiplos sistemas transacionais, com diferentes formas de representação e de tratamento dos dados, a obtenção de um sistema de informação homogêneo é uma tarefa complexa e que tem desde sempre colocado muitos problemas aos arquitetos dos sistemas de apoio à decisão”* (Caldeira, 2012, p. 47).

**Quadro 5 - Sistema de Metadados**

Sistemas de Metadados				
<i>Fontes de dados</i>		<b>Transformação</b>	<i>Data Mart Ocupações</i>	
Origem			Destino	
<i>Tabela</i>	<i>Campo</i>		<i>Tabela</i>	<i>Campo</i>
<b>Sistema SPARES Comissários</b>	Evento	Em todas as etiquetas ocupações, remover as tags inseridas nesse registo	Ocupação	Ocupação
<b>Ficheiro Excel</b>	tipo de estatuto	S/ Transformação	Ocupação	Tipo de Estatuto
<b>Ficheiro Excel</b>	tipo de ocupação	S/ Transformação	Ocupação	Tipo de Ocupação
<b>Ficheiro Excel</b>	tipo de sector	S/ Transformação	Ocupação	Tipo de Sector

O sistema de metadados técnicos do nosso projeto, está desenhado de acordo com o modelo CWM<sup>19</sup>, de forma clara e concisa. No sistema transacional, temos a tabela e o campo

<sup>18</sup> Junções em Português.

<sup>19</sup> “O objetivo do consórcio Common Warehouse Metadata é normalizar o intercâmbio de meta informação entre as ferramentas da data warehousing e os repositórios de meta dados(<http://www.cwmforum.org/>).” (Caldeira, 2012, p. 48).

onde são obtidas as informações para a data mart, e temos a especificação se na extração, tratamento e limpeza dos dados, for ou não necessário efetuar algumas transformações dos dados.

O Quadro 5 apresenta o sistema de metadados técnicos de acordo com o modelo CWM, e teve a finalidade de auxiliar na implementação do data mart ocupações. Através do sistema de metadados podemos compreender as origens dos dados da dimensão ocupação, bem como as transformações que foram necessárias efetuar nos dados que foram carregados no data mart.

Este sistema de metadados técnico na visão de Costa (2012 p. 34) descreve um data warehouse contendo:

- Uma estrutura dos dados;
- As fontes de origem dos dados que o suportam;
- O mapeamento e transformação dos dados à medida que são extraídos, transformados e carregados;
- A relação entre o modelo de dados e o Data Warehouse;
- Os registos de todas as atualizações e carregamentos;
- As definições e/ou descrições dos dados;
- As especificações do modelo de dados.

Os metadados desempenham um papel crucial para um sistema de Data Warehousing, auxiliando na compreensão da estrutura, na definição e significado dos dados de um Data Warehouse, permitindo também entender o que realmente aconteceu e o que está a acontecer no repositório de dados (Costa, 2012, p. 34).

Nas perspectivas da introdução de aplicações de técnicas de Business Intelligence, esta questão coloca-se sobretudo no desenvolvimento de um data warehouse. Sezões, Oliveira, e Baptista (2006, p.40), cita que Ralph Kimball, estudioso e teórico destas áreas, apresentou uma lista dos vários tipos de metadados presentes num data warehouse, para que elas possam ser agregadas da seguinte forma:

- **Metadados de negócio**, elas descrevem o significado dos dados do data warehouse e das ferramentas de business intelligence. Guardam informações importantes como as definições e áreas de negócio, a estrutura e a hierarquia dos dados, as regras de agregação, a definição de métricas de negócio, entre outras;
- **Metadados dos processos**, descrevem a origem, o momento, frequência e forma como foram carregados os dados que estão no data warehouse, ou seja, estão ligados à periodicidade do processo ETL de um determinado data mart;
- **Metadados técnicos**, descrevem os locais físicos, a fonte de dados transacionais, os formatos e os tipos de dados dos elementos de dados, estruturas de ficheiros e tabelas, índices, as suas transformações necessárias, e a fonte de destino, onde serão os dados carregados, conforme exemplo descrito no Quadro 5;
- **Metadados aplicativos**, descrevem a forma como aceder e utilizar os dados. Podem ainda descrever os momentos em que os mesmos são acedidos, por quem (identificam o utilizador) utilizador, com que frequência, etc.

É perceptível que todos os tipos de metadados são fundamentais na gestão dos sistemas de informação das organizações, o que faz com que a tarefa de localizar os dados pretendidos seja simples, e se consiga monitorizar e controlar todo o processo de ETL, desde a fonte de dados às interfaces de análise e reporting de dados. A disseminação de enormes silos de informação, cuja integridade é necessário garantir, torna a gestão dos metadados uma tarefa essencial para otimizar o conhecimento existente nos sistemas de informação de uma organização (Sezões, Oliveira, e Baptista, 2006, p. 41).

Por estas razões, torna-se fundamental ter um repositório de metadados do sistema SPARES. Num projeto futuro, o repositório poderá ser desenvolvido internamente, auxiliando novos utilizadores ou até mesmos novos projetos relacionados com o SPARES. Devido ao facto de não haver um sistema de metadados, no início quando comecei a desenvolver este projeto, senti muitas dificuldades de compreender os dados, logo, se pudéssemos dispor desde logo de um sistema de metadados que fosse utilizado na compreensão dos dados, a interpretação do sistema SPARES seria muito mais facilitada.

Apesar de já ter sido dada uma definição de metadados, é relevante mencionar que de acordo com Sezões, Oliveira, e Baptista (2006, p. 49) “*os metadados sempre estiveram presentes nos sistemas operacionais*”, fazendo parte integrante da documentação dos sistemas, catálogos das bases de dados, etc. Anteriormente, os metadados eram apenas manuseados por pessoas com uma tipologia de conhecimentos eminentemente técnicos, que mantinham os seus próprios sistemas. A evolução da sociedade, fez com que atualmente, no mundo do business intelligence, os metadados não sejam apenas para os técnicos; eles são importantíssimos para os utilizadores finais do sistema, devido ao facto de ser um agente auxiliador na localização dos dados e na sua interpretação dentro do ambiente de business intelligence.

Os mesmos autores, consideram que hoje em dia os metadados assumem um novo papel, que consiste na navegação entre os dados, devendo auxiliar o utilizador a encontrar os dados que necessita dentro do data warehouse.

Sezões, Oliveira, e Baptista (2006, p. 49) indicam algumas condicionantes necessárias para o desenvolvimento de um sistema de metadados. Para a sua concepção devemos dar respostas às seguintes perguntas:

- Que tipos de metadados de negócios são necessários?
- Que tipos de metadados técnicos são necessários?
- Quem será responsável pelo sistema de metadados?
- Quem e como se terá acesso ao repositório de metadados?
- Servirá o repositório de metadados apenas as aplicações de BI e o DW ou toda a organização?

# Capítulo 5

## **Casos de Estudios**



## 5. Casos de Estudo

É inegável a importância do presente estudo, pois a aplicabilidade das múltiplas técnicas e métodos de Business Intelligence a uma rede prosopográfica, revela-nos informação histórica sobre eventos biográficos, relacionais e de parentesco entre diversos protagonistas históricos (Ferreira, Caldeira, e Olival, 2012). É objetivo central, que este projeto possa ser uma componente útil e relevante para o CIDEHUS, que proporcione inclusive aos seus utilizadores a descoberta de padrões de dados insólitos, oferecendo em suma maior flexibilidade na exploração e análise dos dados. O desenvolvimento de uma aplicação de análise de dados de tipo OLAP, vai contribuir de modo significativo para o desenvolvimento do projeto. Entre outras componentes do objeto sobre o qual versa o estudo, podemos referir a criação de data marts sobre ocupação, permitindo descobertas de novos padrões de dados relacionados com a ocupação.

Outro campo de estudo foi a criação de chaves artificiais, que é um campo que serve de chave primária numa dimensão e, *uma vez que o sistema SPARES não as possui, será possível criá-las?*. Outra é a dimensão lixo<sup>20</sup>, *que é útil porque fornece informação adicional sobre um acontecimento*. Refere ainda Caldeira que, *“as características principais destes atributos são a descontextualização, pelo facto de não encaixarem em nenhuma dimensão.”* (Caldeira, 2012, p. 145). Sendo assim, vamos analisar se é possível criar uma dimensão lixo no sistema SPARES, tornando as dimensões independentes da sobrecarga de atributos caracterizadores, mas que, a um primeiro nível, não são relevantes para o processo de análise de dados. O objeto primordial do estudo é a otimização de análise temporal, que consiste na criação de uma dimensão Data, dentro de um intervalo de tempo entre o século XVI - século XIX. Depois deverá criar-se e estudar a possibilidade da sua integração no sistema SPARES, dado que o modo de efetuar os registos das datas são de formas ambíguas, pelo facto da maioria das vezes não conhecermos a data concreta, o faz com que ela seja registada como um tipo de texto, e não com o tipo de data.

Para complementar os casos de estudo, utilizaremos essas transformações no sistema SPARES, para facilitar a aplicação de técnicas de Business Intelligence a um sistema prosopográfico de análise de relações e eventos. Com o aplicativo escolhido, efetuar as análises de dados é muito pratico e simples, análises essas que se forem realizadas com as ferramentas atuais de pesquisas de dados, podemos considerar que à partida é impossível, mas com as novas alterações no sistema e com o Tableau temos ferramentas potentes para explorar os dados referentes à atuação de comissários no Santo Ofício. É de salientar que a princípio o caso de estudo centralizava-se na criação e na inserção de análise temporal no sistema SPARES, mas ao longo do projeto, surgiram novas ideias e casos de estudo de grande interesse para os investigadores e utilizadores do sistema SPARES.

Nomeadamente, a análise sobre as ocupações dos comissários e suas respetivas genealogias, tendo desta forma o objeto de estudo tomado novas diretrizes, a fim de resolver as demandas dos utilizadores e oferecer novas faces de pesquisa ao sistema SPARES.

---

<sup>20</sup> **Junk Dimension**, em inglês.

## 5.1. O Sistema SPARES

A base de dados prosopográfica – o sistema SPARES – que é um repositório de dados com um suporte em informações prosopográficas, auxiliando o CIDEHUS no arquivo de informações sobre a época da Inquisição, não tem ferramentas adequadas e flexíveis para as pesquisas e análise de dados, e descobertas de novos padrões de dados. A data warehouse é um método eficaz para a resolução desse tipo de problemas. Já existe a data warehouse que armazena os dados prosopográficos juntamente com filtros de pesquisas, que são ferramentas demasiado básicas para pesquisa e descoberta de padrões de dados.

O sistema prosopográfico de análise de relações e eventos desempenha duas funções: é um sistema transacional e também é um sistema analítico. O SPARES serve para registrar as transações sobre as relações e eventos relacionados com a atuação dos comissários do Santo Ofício, contudo, tem também a serventia para análise dos respetivos dados.

Este facto, faz com que tenha maior grau de dificuldade em analisar os dados, tendo em conta que é um repositório de armazenamento de dados desnormalizado e ao mesmo tempo desempenha a função analítica. Há nove anos atrás quando o modelo de dados do SPARES foi desenhado conforme a Figura 19, o mesmo foi pensado para desempenhar o papel de um data warehouse.

Observando a Figura 20, onde consta o modelo de dados do SPARES inicial, já com a introdução da dimensão Data, podemos afirmar que este sistema está dividido em duas áreas. Uma delas é a área de tabela de factos comissários, que está ligada a diversas dimensões, onde podemos pesquisar e explorar dados sobre as atuações dos comissários do Santo Ofício. A outra vertente, é pesquisar sobre genealogias, o que dificulta as pesquisas importantes aos utilizadores e investigadores, dado que são perguntas que de certa forma juntam as duas vertentes, mas que não tem nenhuma ligação simplificadora para se efetuar a pesquisa.

A título de exemplo podemos mencionar, uma análise para identificar concelhos com mais ocupações mecânicas a cada 25 anos, a começar em 1575, e que sejam familiares do Santo Ofício, ou concelhos com mais alfaiates que sejam familiares Santo Ofício, ou ainda, analisar concelhos com mais familiares oriundos de outros concelhos, ou seja, que não nasceram no mesmo concelho onde moram.

Dado que o sistema SPARES é complexo de gerir, processar e analisar os grandes volumes de dados armazenados, este projeto está focado especificamente na aplicabilidade de técnicas de Business Intelligence sobre os dados prosopográficos. Do mesmo modo, o projeto terá como objetivo ser um instrumento facilitador para as necessidades dos investigadores, e utilizadores desse sistema, contribuindo assim para o aprimoramento do processo decisório em questões prosopográficas, e descoberta de novos padrões de dados existentes.

Sendo o SPARES um data warehouse, a classe de ferramentas mais indicada para proceder à análise dos seus dados são as aplicações OLAP que são as melhor posicionadas para poderem explorar os dados sob diferentes perspectivas.

Figura 19 - Modelo dimensional do SPARES

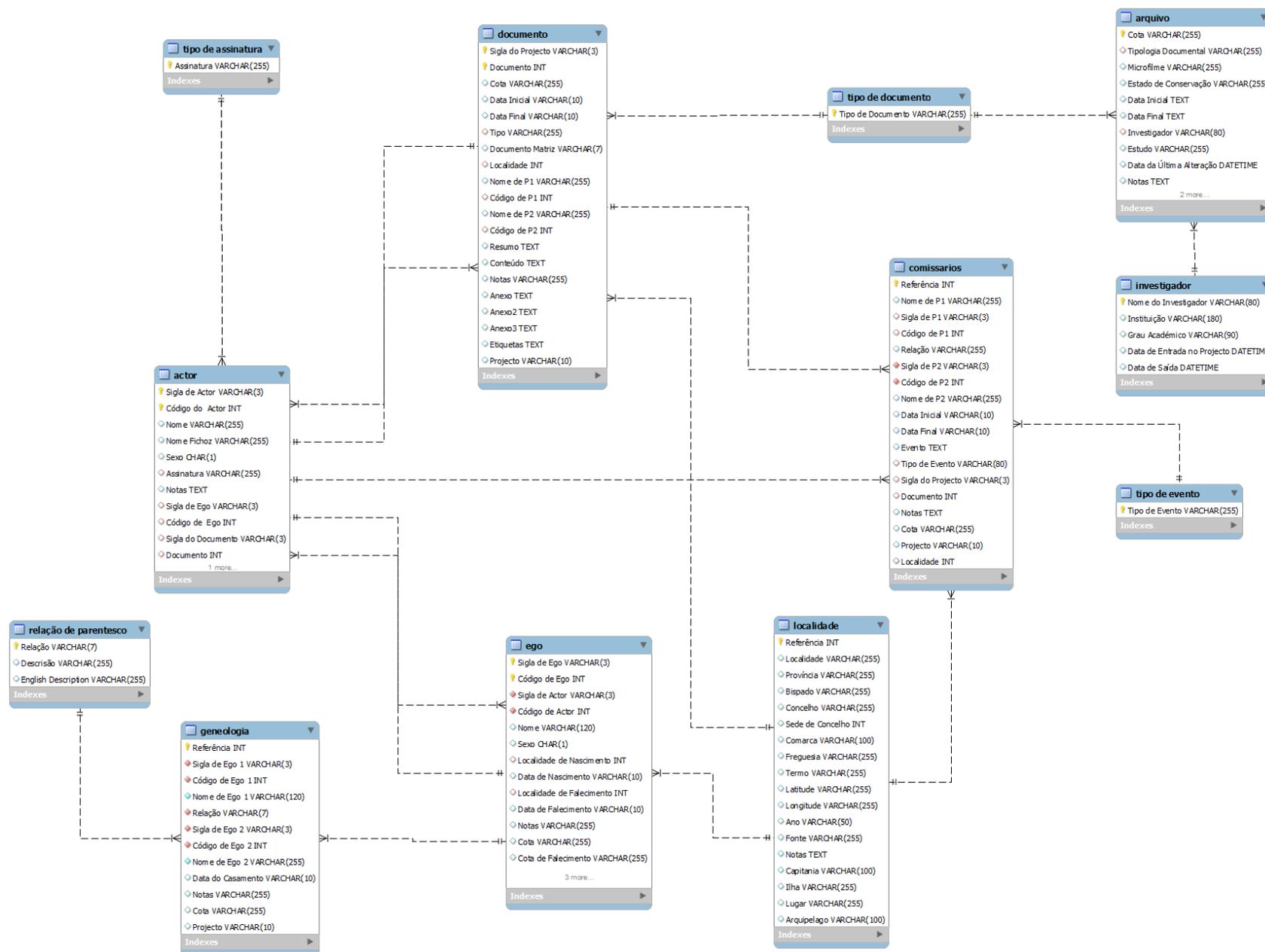
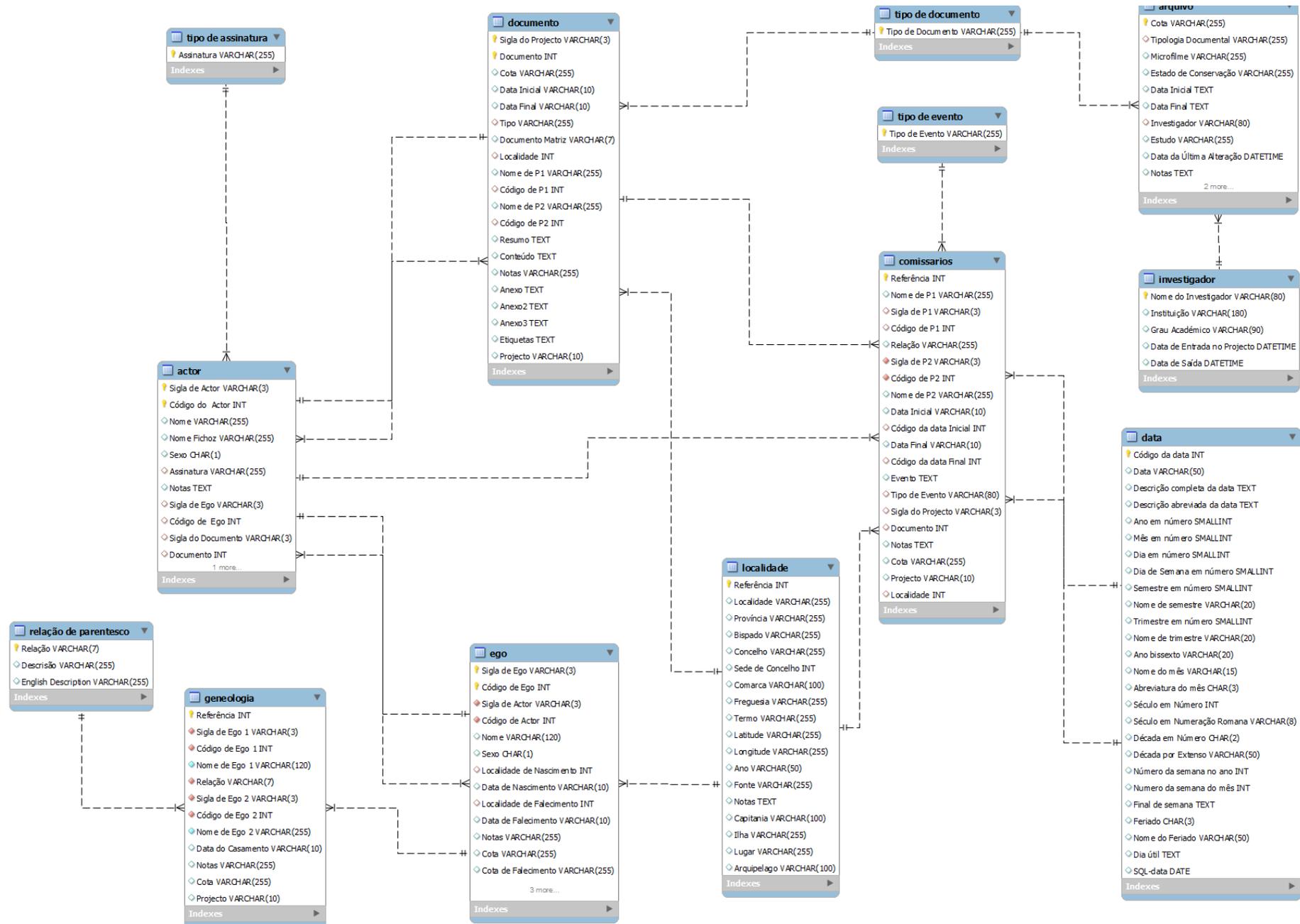


Figura 20 - Molelo dimensional do SPARES com a nova dimensão temporal.



### 5.1.1. Metodologia atual de exploração dos dados

Atualmente a forma de explorar os dados no sistema prosopográfico de análise de relações e eventos é através de filtros avulsos. Como podemos observar na Figura 21, é uma ferramenta com muito pouca potência para analisar os dados. Não permite realizar agregações de dados, não permite realizar consultas complexas onde constam relações entre tabelas, só permitindo realizar segmentações básicas.

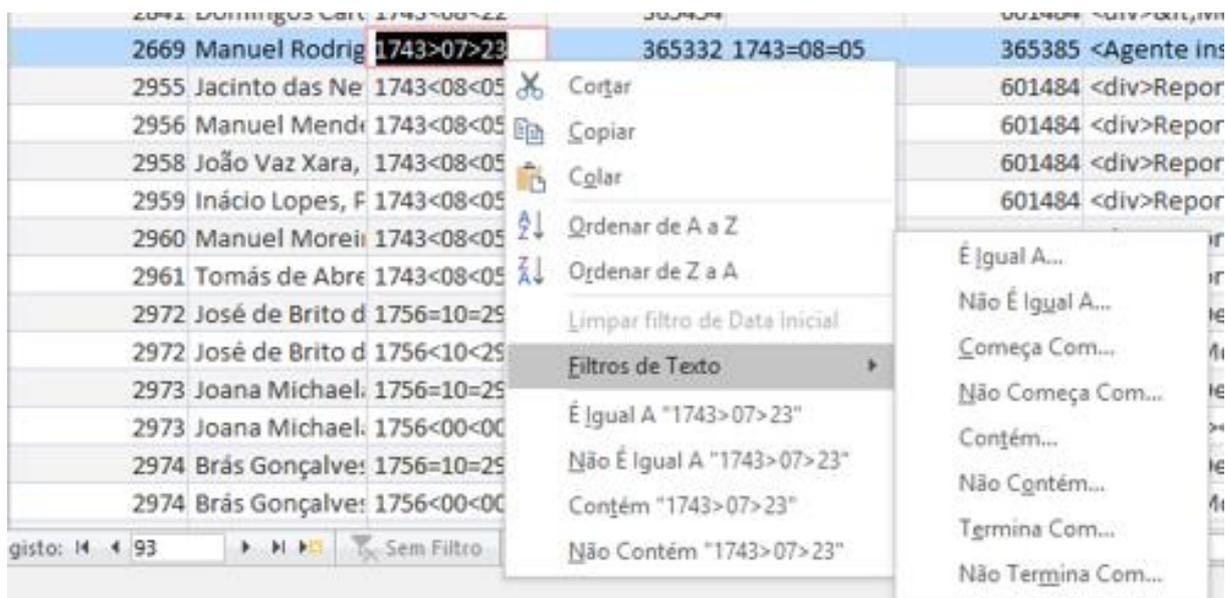
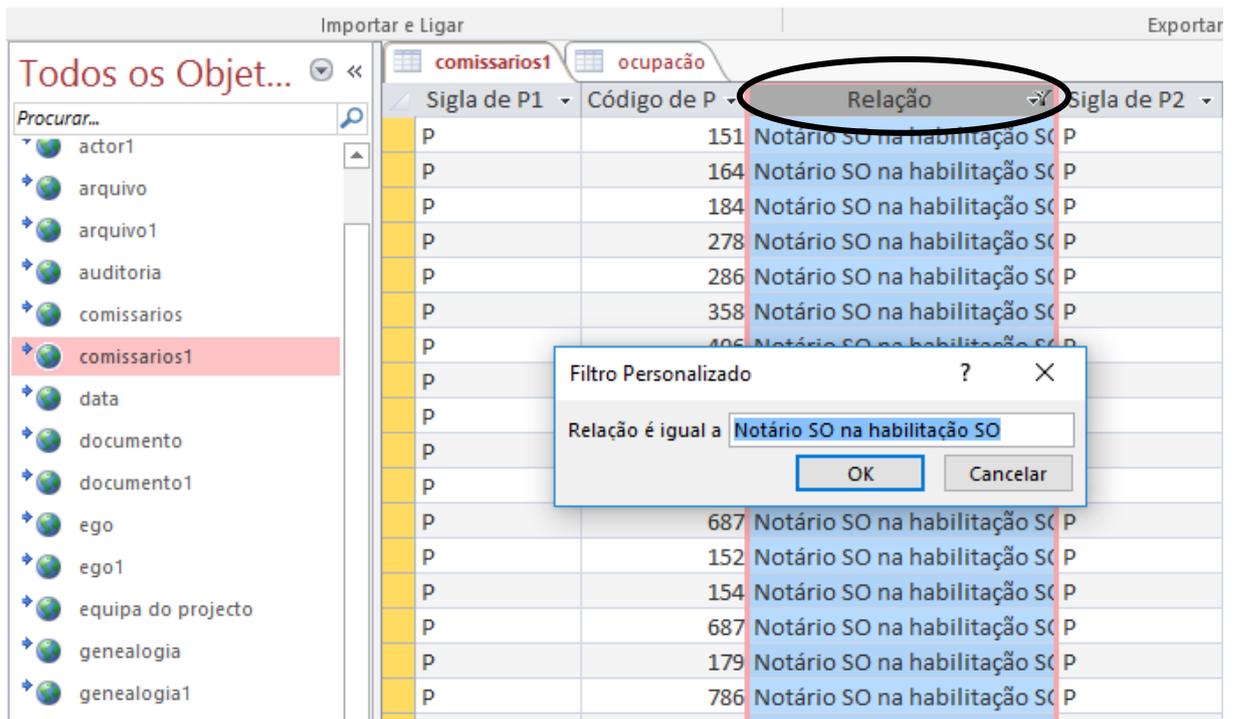


Figura 21 - Exemplo 1 de forma atual de exploração de dados no SPARES

Os filtros são uma forma útil de ver somente os dados que o utilizador deseja ver. É possível usar os filtros para exibir registos específicos num formulário, um relatório, uma consulta, ou para mostrar somente determinados registos de um relatório, uma tabela ou uma consulta. Com a aplicação de um filtro, o utilizador pode limitar os dados numa exibição sem alterar o design do objeto.

Podemos constatar na Figura 21 a exibição obtida depois que foi aplicado o filtro, verificando-se que o resultado contém apenas registos com os valores selecionados, e o restante dos dados permanecerá oculto até o utilizador efetuar a limpeza do filtro. Existem vários tipos de filtros, e normalmente são muito fáceis de aplicar e remover sobre um conjunto de dados. A disponibilidade dos comandos de filtros depende do tipo e dos valores do campo, pelo que não podemos aplicar um só filtro para diversas colunas.

Na Figura 21 o tipo de dados do campo selecionado é uma sequência de caracteres, e se observarmos bem, vemos que as opções que temos para o filtro são: filtrar o campo com a opção «É igual A...» um valor ou simétrico disso, filtrar o campo com textos que «Começa com...», ou «Termina com...»; uma das opções com uma maior usabilidade é a opção de filtrar o campo com «Contém...» ou «Não Contém...» um determinado valor quer inteiro ou texto.



**Figura 22 - Exemplo 2 de forma atual de exploração de dados no SPARES**

Os filtros disponíveis dependem do tipo de dados na coluna selecionada. No momento que o utilizador aplica um filtro a uma coluna que já foi filtrada, o filtro anterior remove-se automaticamente, pelo facto de que somente um filtro por vez pode estar em vigor para um determinado campo, mas possibilita definir um outro filtro num campo que não esteja filtrado.

Observando a Figura 22 o ícone de filtro no cabeçalho da coluna indica que o modo atual está filtrado pela coluna «Relação» onde os dados é igual a «Notário SO na habilitação SO»; se o utilizador preferir, a posteriori poderá imprimir o relatório apenas com esses dados filtrados.

Os filtros não são nada mais, nada menos do que um simples «WHERE» no comando SQL. Tem como suas vantagens a possibilidade de ser considerada para uma consulta rápida, mas não como uma ferramenta para pesquisas e análises de dados, pelo facto de não ter capacidades de descobrir novos padrões de dados, e não é uma ferramenta para auxiliar os gestores nas tomadas de decisões, dado que é impossível gerar informações a partir dos dados que possam ser úteis pra tomadas de decisão.

Foi esta factualidade que fez com que nascesse esse projeto de estudo, de melhorar a forma atual de análise de dados no sistema SPARES. É então, nesse contexto, que surgem as técnicas e ferramentas de Business intelligence para resolver o grande problema do sistema SPARES, que é oferecer outra face ao sistema no âmbito de análise e exploração de dados. Todas as dificuldades que os investigadores tinham para analisarem os dados e descobrirem novos padrões de dados, foi substancialmente melhorado com o aplicativo tipo cubo, pois torna-se tudo muito intuitivo e oferece maior qualidade de dados nas suas análises e apresentação de reporting de dados com diversos tipos de visualização.

### 5.1.2. As Tabelas dimensões do Sistema SPARES

O cubo OLAP sobre o qual vão ser realizadas as análises de dados, é constituído por uma tabela de factos central ligado a diversas dimensões. As dimensões podem ter diversas naturezas, podendo ser uma dimensão ponte, uma mini dimensão, uma dimensão degenerada, entre outras. Algumas dimensões, permitem ao utilizador segmentar os dados em hierarquias, oferecendo várias perspetivas de análise de dados. Podemos classificar as dimensões do sistema SPARES em três tipos de dimensões, que são as seguintes:

- **Dimensões auxiliares**, são dimensões que têm como objetivo primordial servir de auxílio à tabela de factos e também a outras dimensões, que são as seguintes:

Quadro 6 - A dimensão Relação de Parentesco

Relação de Parentesco	
PK	Relação
	Descrição English Description

Quadro 7 - A dimensão Tipo de Evento

Tipo de Evento	
PK	Tipo de evento

Quadro 8 - A dimensão Tipo de Assinatura

Tipo de Assinatura	
PK	Assinatura

Quadro 9 - A dimensão Investigador

Investigador	
PK	Nome do Investigador
	Instituição Grau Académico Data de Entrada no Projecto Data de Saída

Quadro 10 - A dimensão Tipo de Documento

Tipo de Documento	
PK	Tipologia documental

- **Dimensões preparatórias**, são as dimensões adjuvantes de tabela de factos, e reúnem informações importantes, destacando-se as seguintes:

**Quadro 11 - A dimensão Ego**

Ego	
PK	<u>Código do medicamento</u>
PK	<u>Sigla de Ego</u>
	Código de Ego
	Sigla de Actor
	Código do Actor
	Nome
	Sexo
	Localidade de Nascimento
	Data de Nascimento
	Localidade de Falecimento
	Data de Falecimento
	Notas
	Cota
	Cota de Falecimento
	Ocupação
	Cota da Ocupação
	Projecto

**Quadro 12 - A dimensão Ator**

Actor	
PK	<u>Sigla de Ator</u>
PK	<u>Código do Ator</u>
	Nome
	Nome Fichoz
	Sexo
	Assinatura
	Notas
	Sigla de Ego
	Código de Ego
	Sigla do Documento
	Documento
	Projecto

O Quadro 12 apresenta a dimensão ator que tem como função importante a de registar a pessoa ou o indivíduo que pratica ou está envolvido numa ação ou evento, sendo que a grafia utilizada para o campo nome é a grafia atualizada.

A dimensão representada no Quadro 13 é de muita importância para os investigadores, pelo facto do resultado das análises de dados na maioria das vezes ser cartografado em mapas de Portugal de acordo com o ano 1527-1532. Essa dimensão valoriza o pequeno topónimo, como por exemplo o lugar do acontecimento, tendo também entradas genéricas como Madeira, Lisboa, entre outras. Possui um surrogate keys para quando não se sabe ao certo o local, e os utilizadores usam a grafia atual nos registos de dados, mas não usam o topónimo atual.

Quadro 13 - A dimensão Localidade

Localidade	
PK	Referência
	Localidade
	Província
	Bispado
	Concelho
	Sede de Concelho
	Comarca
	Freguesia
	Termo
	Latitude
	Longitude
	Ano
	Fonte
	Notas
	Capitania
	Ilha
	Lugar
	Arquipelago

Quadro 14- A dimensão Arquivo

Arquivo	
PK	Cota
	Tipologia Documental
	Microfilme
	Estado de Conservação
	Data Inicial
	Data Final
	Investigador
	Estudo
	Data da Última Alteração
	Notas
	Data de Início do Trabalho
	Projecto

- **Dimensões nucleares**, as dimensões principais e a tabela de factos.

**Quadro 15 - A dimensão Geneologia**

Geneologia	
PK	Referência
	Sigla de Ego 1
	Código de Ego 1
	Nome de Ego 1
	Relação
	Sigla de Ego 2
	Código de Ego 2
	Nome de Ego 2
	Data do Casamento
	Notas
	Cota
	Projeto

**Quadro 16 - A dimensão Documento**

Documento	
PK	Sigla do Projecto
	Documento
	Cota
	Data Inicial
	Data Final
	Tipo
	Documento Matriz
	Localidade
	Nome de P1
	Código de P1
	Nome de P2
	Código de P2
	Resumo
	Conteúdo
	Notas
	Anexo
	Anexo2
	Anexo3
	Etiquetas
	Projecto

A dimensão documento representado no Quadro 16, destina-se a fornecer o contexto da informação. Pode ter sub-documentos ou documentos filhos.

Sendo o SPARES um repositório de dados com suporte informacional prosopográfico, podemos afirmar que está dividido em dois repositórios. Um repositório de análise e classificações de relações e eventos e outro de análise de genealogia. No repositório principal, que é a de análises de relações e eventos, foram adicionadas mais dimensões ao modelo de dados do sistema SPARES. Nomeadamente, a dimensão Data que encontra explicada no capítulo 5.2, a sua concepção, desenvolvimento e implementação, e a dimensão ocupação que está demonstrada no Quadro 17, e explicada com maior detalhe a sua implementação no capítulo 4.7.

Quadro 17 - A dimensão Ocupação - nova dimensão no SPARES

Ocupação	
<b>PK</b>	<b>Código de Ocupação</b>
	Ocupação
	Tipo de Sector
	Tipo de Estatuto
	Tipo de Ocupação

### 5.1.3. A Tabela de Facto do Sistema SPARES

Quadro 18 - Tabela de Factos «comissários»

Comissarios	
<b>PK</b>	<b>Referência</b>
	Nome de P1
	Sigla de P1
	Código de P1
	Relação
	Sigla de P2
	Código de P2
	Nome de P2
	Data Inicial
	<b>Código da Data Inicial</b>
	Data Final
	<b>Código da Data Final</b>
	Evento
	<b>Ocupação</b>
	Tipo de Evento
	Sigla do Projecto
	Documento
	Notas
	Cota
	Projecto

A Figura 18 demonstra que a tabela de factos constitui o centro do esquema em estrela, armazenando enormes quantidades de informação. Assim, cada registo de uma tabela de factos possui uma coluna, ou um conjunto de colunas, que corresponde às chaves primárias de cada uma das tabelas dimensionais do esquema em estrela. Além destas colunas com as chaves estrangeiras, a tabela de factos contém também colunas que descrevem medidas numéricas que possam ser agregadas (através de somas, contagens, médias, etc.).

Isto significa que, a tabela de factos diferente das dimensões é altamente normalizada, pois a sua chave primária consiste na concatenação de todas as chaves estrangeiras das dimensões relevantes para os factos, e os atributos são medidas numéricas que se relacionam diretamente com a chave primária.

## 5.2. A dimensão Data

De acordo com Caldeira (2012, p.110), “As *datas* são uma grandeza fundamental em todos os negócios e organizações. (...) A dimensão Data vai servir de exemplo para o estudo pormenorizado da estrutura hierárquica dos dados que se encontra frequentemente no interior das dimensões. O desenho desta dimensão tem que ser devidamente avaliado de acordo com as utilizações previstas.”

A dimensão Data é na prática a mais utilizada das dimensões usadas na maioria dos sistemas de Business Intelligence. Prova disso, encontramos-na na citação de Caldeira (2012, p.110) (.), de acordo com o qual “*Em todos os tipos de data marts é comum a presença de uma dimensão data composta por múltiplos atributos que descrevem cada uma das ocorrências da tabela*”. Apesar de ter a mesma analogia em todos os sistemas, não é composta pelos mesmos atributos, pois elas podem ser adaptadas de acordo com a necessidade do projeto envolvente.

Quadro 19 - A dimensão Data

Data		Exemplo
<b>PK</b>	<b>Código da data</b>	<b>1</b>
	Data	21/09/2016
	Descrição completa da data	21 de Setembro de 2016
	Descrição abreviada da data	21 de Set de 2016
	Ano em número	2016
	Mês em número	9
	Dia em número	21
	Dia de Semana em número	4
	Semestre em número	2
	Nome de semestre	Segundo Semestre
	Trimestre em número	3
	Nome de trimestre	Terceiro Trimestre
	Ano bissexto	É ano bissexto
	Nome do mês	Setembro
	Abreviatura do mês	Set
	Nome de dia	Quarta Feira
	Abreviatura do dia	Qua
	Século em Número	21
	Século em Numeração Romana	XXI
	Década em Número	20
	Década por Extenso	Década 20 do Século XXI
	Número da semana no ano	43
	Numero da semana do mês	3
	Estação do ano	Verão
	Final de semana	Não é final de semana
	Feriado	Não É Feriado
	Nome do Feriado	Null
	Dia útil	É dia útil
	SQL-data	21/09/2016

O encaixe de uma dimensão Data no modelo dimensional é um dos casos do estudo deste projeto, dado que a sua existência é de grande relevância. O desafio centra-se em dar resposta à seguinte questão: **Será possível a concepção da dimensão Data e o seu enquadramento no sistema de Business Intelligence SPARES orientado para a análise prosopográfica do Santo Ofício entre os séculos XVI e XVIII?**

No caso de ser confirmada essa possibilidade, trará grandes vantagens ao modelo de dados dimensional, fazendo com que seja extremamente simples, intuitivo e flexível analisar os dados de acordo com os segmentos temporais, o que permitirá aos utilizadores descobrirem e identificarem mais facilmente novos padrões de informação. A conceção dessa dimensão foi adaptada consoante a natureza do sistema SPARES.

Fisicamente criou-se uma tabela, denominada de “data”. A sua chave primária é o atributo “Código da Data”, que é um campo numérico e automático sem nenhum significado, a sua granularidade<sup>21</sup> é o dia, ou seja, os factos são agregados por dia. Não podemos deixar de referir que é caracterizada pela sua desnormalização, possibilitando, apesar disso, um melhor desempenho na análise efetuada pelos utilizadores (Microsoft, 2010).

No Quadro 19 podemos constatar o exemplo da dimensão Data deste projeto e respetivos atributos.

A dimensão Data contém diversos atributos. Podemos citar o campo de Descrição Completa da Data “Quarta-Feira, 21 de Setembro de 2016”, e também pode ser definida de forma abreviada “Qua, 21 de Set de 2016”. Segundo Caldeira (2012, p.111) “Para cada um dos dias é possível ainda acrescentar uma série de indicadores”, e de acordo com a indicação adicionamos os seguintes atributos: O Nome de Dia, na forma «Domingo, ...Sábado»; a Abreviatura do Dia na forma «Dom, ...Sab»; O Dia da Semana em número começando em «1, ...7» sabendo que 1 corresponde ao domingo e 7 a sábado; o Nome do Semestre e Trimestre em número e por extenso; O Ano em Número e o Ano se é um Ano Bissexto; O Nome do Mês « Janeiro, ...Dezembro», e suas respetivas abreviaturas «Jan, ...Dez»; Se é Final de Semana; Se é Feriado e o Nome do Feriado; Se é Dia Útil. Estes últimos indicadores são exemplos de atributos a que se poderia responder com um “S”, “Sim” e “N”, “Não” . Portanto, de acordo com Caldeira (2012, p.112)” para manter o máximo de legibilidade no data warehouse, os valores nas colunas têm que ser sempre explícitos; no caso do indicador de fim de semana: «É fim de semana» ou, caso contrário, «Não é fim de semana».

Como o sistema SPARES tem como fonte os dados prosopográficos, consideramos que os casos de estudo são de séculos passados, tendo um atributo de Século e Década em número e por extenso. Existem igualmente outros indicadores particulares, como a estação do ano, caracterizado como «verão, inverno, outono e primavera».

Nos sistemas transacionais, segundo Caldeira (2012, p.113), “as datas são comumente registadas em campos cujo domínio é um tipo especial denominado SQL-data. Essas datas são a base para todos os cálculos cujos, através do sistema ETL, (...) valores

<sup>21</sup> “Nível de detalhe com que os dados são medidos e armazenados.

Quanto maior a granularidade menor o nível de detalhe e vice-versa” (Caldeira, 2012, p. 207).

SQL-data vão passar para o data warehouse para um atributo na dimensão Data,” com o formato de «date».

### 5.3. Aplicação da dimensão Data a um ambiente prosopográfico

No sistema de BI, as datas são geralmente conhecidas. Exemplo disso, é o que acontece no supermercado com a venda de um produto, processada numa data exata. Mas o caso da prosopografia é especial, dado que nem sempre conhecemos a data exata.

A título de exemplo podemos referir, o caso em que não se sabe a data concreta, mas sabe-se que o evento aconteceu após essa data. Essa dimensão trará grandes vantagens ao modelo dimensional, tornando-o extremamente simples, intuitivo e flexível, uma vez que permite analisar os dados de acordo com os segmentos temporais.

No caso do sistema SPARES, a dimensão Data, como de modo similar na grande maioria dos casos, não indica a data concreta do evento, tendo então sido criado um sistema de data da seguinte forma:

- Quando a data for apresentada sob a forma “2016=09=21”, sabe-se que corresponde à data exata;
- Quando a data for apresentada sob a forma “2016<09<21”, não se sabe a data concreta, mas sabe-se que o acontecimento ocorreu antes dessa data;
- Quando a data for apresentada sob a forma “2016<09<00”, não se sabe o dia concreto, nem a data concreta, mas sabe-se que o evento ocorreu antes de setembro de 2016;
- Quando a data for apresentada sob a forma “2016<00<00”, não se sabe o dia nem o mês concreto, nem a data concreta, mas sabe-se que o evento ocorreu antes do ano de 2016.
- Quando a data for apresentada sob a forma “2016>09>21”, não se sabe a data concreta, mas sabe-se que o acontecimento ocorreu depois dessa data;
- Quando a data for apresentada sob a forma “2016>09>00”, não se sabe o dia concreto, nem a data concreta, mas sabe-se que o evento ocorreu depois de setembro de 2016;
- Quando a data for apresentada sob a forma “2016>00>00”, não se sabe o dia nem o mês concreto, nem a data concreta, mas sabe-se que o evento ocorreu depois do ano de 2016.
- Quando a data for apresentada sob a forma “2016-09-21”, não se sabe a data concreta, mas sabe-se que o acontecimento ocorreu por volta dessa data;
- Quando a data for apresentada sob a forma “2016-09-00”, não se sabe o dia concreto, nem a data concreta, mas sabe-se que o evento ocorreu por volta de setembro de 2016;
- Quando a data for apresentada sob a forma “2016-00-00”, não se sabe o dia nem o mês concreto, nem a data concreta, mas sabe-se que o evento ocorreu por volta do ano de 2016.

Por esta razão, o desenho da dimensão Data foi devidamente avaliado e projetado. Após a construção física da dimensão, foi programada uma função descrita no Anexo B, que recebe dois argumentos formais (data inicial e data final). Portanto, efetua o cálculo entre esse intervalo de datas, incrementando o grão da dimensão Data, e em seguida efetua a inserção na dimensão Data. A título de exemplo, utilizamos um intervalo de datas entre '1500/01/01' e '1900/12/31'.

No Quadro 20 indicam-se alguns exemplos de dados referentes a data. Primeiro regista a data concreta preenchendo todos os campos; a seguir regista a mesma data, mas com um pormenor diferente, pois na descrição completa da data é introduzido “Sabe-se que é antes, 1 de janeiro de 1500”; a utilidade desse registo é verificada no momento em que não sabemos a data concreta, mas sabemos que o evento se realizou antes dessa data. A outra condição é verificar sempre que é início do ano, e introduz um registo “Sabe-se que é antes de 1500”, e por último, no início de cada mês é inserido um registo “Sabe-se que é antes de, Janeiro de 1500”, conseqüentemente, por esse motivo, nem todos os atributos da data poderão ser completados, uma vez que é desconhecido o dia e o mês, ou é desconhecido o dia.

**Quadro 20 - Vista parcial de dados na dimensão Data**

Código da Data	Data	Descrição da Data	Dia em Número	Mês em Número	Nome de Semestre	Ano Bissexto	Estação do Ano	SQL-Data
1	1500-01-01	Segunda-Feira, 1 de Janeiro de 1500	1	1	Primeiro Semestre	É Ano Bissexto	Inverno	1500-01-01
2	1500-01-01	Sabe-se que é antes de, 1 de Janeiro de 1500	1	1	Primeiro Semestre	É Ano Bissexto	Inverno	1500-01-01
3	1500-01-01	Sabe-se que é depois de, 1 de Janeiro de 1500	1	1	Primeiro Semestre	É Ano Bissexto	Inverno	1500-01-01
4	1500-01-01	Sabe-se que é por volta de, 1 de Janeiro de 1500	1	1	Primeiro Semestre	É Ano Bissexto	Inverno	1500-01-01
5	1500-01-00	Sabe-se que é antes de, Janeiro de 1500	NULL	1	Primeiro Semestre	É Ano Bissexto	NULL	NULL
6	1500-01-00	Sabe-se que é depois de, Janeiro de 1500	NULL	1	Primeiro Semestre	É Ano Bissexto	NULL	NULL
7	1500-01-00	Sabe-se que é por volta de, Janeiro de 1500	NULL	1	Primeiro Semestre	É Ano Bissexto	NULL	NULL
8	1500-00-00	Sabe-se que é antes do Ano, 1500	NULL	NULL	NULL	É Ano Bissexto	NULL	NULL
9	1500-00-00	Sabe-se que é depois do Ano, 1500	NULL	NULL	NULL	É Ano Bissexto	NULL	NULL
10	1500-00-00	Sabe-se que é por volta do Ano, 1500	NULL	NULL	NULL	É Ano Bissexto	NULL	NULL

É de realçar o desenvolvimento do script de criação, e inserção dos dados na dimensão Data, dado que a priori foi desenvolvido no SQL-Server. Contudo, pelo facto do sistema SPARES ter sido desenvolvido no MySQL por várias razões, dentre as quais podemos citar o facto de ser *open source*, a sua compatibilidade suporta praticamente qualquer plataforma, e é gratuito. Sendo assim, foi inevitável converter os scripts de Microsoft SQL Server, para MySQL, obtendo deste modo a ubiquidade na criação da dimensão data.

#### 5.4. Metodologias aplicada na Inserção de dados na dimensão Data

Para estabelecer a ligação entre a dimensão Data e a tabela de facto comissários, que segundo Caldeira (2012, p.67), “*é o elemento principal no modelo dimensional*”, foi necessário criar dois atributos (Código da Data Inicial e Código da Data Final) na tabela de facto, do tipo inteiro, que são chaves estrangeiras<sup>22</sup> ligadas com a dimensão Data. A existência dessas duas novas colunas na tabela de facto, deriva do facto de que cada relação ou evento

<sup>22</sup> “As **Chaves estrangeiras**, para além de estabelecerem regras de integridade referencial com as tabelas de dimensão, utilizam as dimensões para caracterizar as métricas registadas e para parametrizar as queries” Caldeira (2012, p. 67).

tem uma data de início e uma data de términos, mas na maioria das vezes uma ou outra é desconhecida. Por esta razão, existe uma *surrogate key* que resolve os casos em que não existe um valor para as datas, quer seja inicial ou final, como podemos constatar na Figura 23.

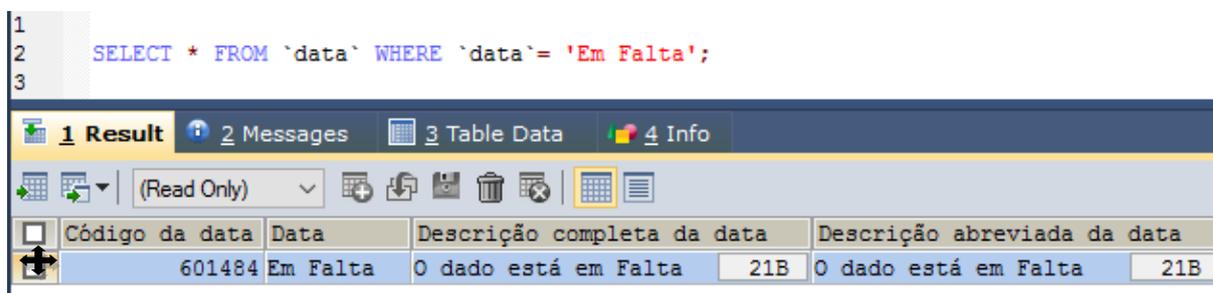


Figura 23 - A utilização de surrogate key na dimensão Data

Este surrogate keys indicado na Figura 23, soluciona os eventos onde não há um valor conhecido da data. Contudo, podemos observar a chave primária da dimensão Data que é homogéneo das chaves das outras dimensões, e correspondente à utilização de chave artificial. De acordo com Caldeira (2012, p.116) “*diferente das outras dimensões, a dimensão data tem uma particularidade, que é a possibilidade da chave artificial poder ter um significado concreto*”, mas de acordo com o nosso caso de estudo tal facto não seria possível, dado que podemos obter conflitos com a integração dos dados, devido à necessidade de registar a mesma data quatro vezes, unicamente com a diferença que se observa Figura 24.

O primeiro registo (Código da data = 1) representa a data concreta, ou seja, o evento aconteceu nessa data; o segundo registo (Código da data = 2) corresponde a um evento que não se sabe a data exata do acontecimento, mas temos o conhecimento que tal evento aconteceu antes dessa data; o terceiro registo (Código da data = 3), da mesma forma que o segundo registo, não tem data conhecida, mas sabe-se que o tal evento aconteceu depois dessa data, e o quarto registo (Código da data = 4), sabe-se que o evento aconteceu por volta dessa data, mas não se sabe se foi antes ou depois da data.

Código da data	Data	Descrição completa da data	Descrição abreviada da data
1	1500-01-01	Segunda-Feira, 1 de Janeiro de 1500	35B Seg, 1 de Jan de 1500
2	1500-01-01	Sabe-se que é antes de, 1 de Janeiro de 1500	45B Sabe-se que é antes de, 1 de Jan de 1500
3	1500-01-01	Sabe-se que é depois de, 1 de Janeiro de 1500	46B Sabe-se que é depois de, 1 de Jan de 1500
4	1500-01-01	Sabe-se que é por volta de, 1 de Janeiro de 1500	49B Sabe-se que é por volta de, 1 de Jan de 1500

Figura 24 - Exemplo de registos para a dimensão Data

As segmentações são uma das formas mais eficientes do aplicativo tipo cubo apresentar a visualização dos dados. Uma segmentação, é um filtro de elemento visual que permite a qualquer utilizador que observe o relatório, segmentar os dados por determinado valor, seja por ano, semestre, trimestre, dia, estação do ano ou por localização geográfica. Nesse aspeto, a dimensão data traz uma nova face ao sistema SPARES, proporcionando as análises de relações e eventos através do horizonte temporal.

Segundo Caldeira (2012, p. 138), “*no desenvolvimento de um modelo dimensional surgem frequentemente múltiplas datas que funcionam como marcas temporais importantes em quase todos os processos de negócio de uma organização*”. É o caso presente no sistema

prosopográfico de análise de relações e eventos, sendo a data neste sistema tratada de forma diferente de outros sistemas.

As datas do acontecimento são colocadas na tabela de factos e funcionam como chaves estrangeiras da dimensão, ao contrário das demais datas que são simplesmente registadas nas dimensões como um campo.

# Capítulo 6

## **Análise de Dados**



## 6. Análise de Dados

### 6.1. Ferramenta para Análise dos dados.

A ferramenta escolhida para a análise de dados no sistema SPARES, é o Tableau. É considerada como sendo uma solução que auxilia os seus utilizadores a visualizarem e compreenderem os dados, de uma forma simples e rápida.

O Tableau permite que os utilizadores do SPARES transformem milhares de dados em informações úteis. Com essa ferramenta, vão ter a possibilidade de melhorar a forma atual de análise de dados no sistema SPARES e explorar os seus dados através de uma análise visual muito dinâmica.

### 6.2. Sobre o Tableau

Sob a perspetiva histórica, tudo terá começado com três homens de Stanford, um cientista<sup>23</sup> da computação, um professor<sup>24</sup> premiado com um Oscar e um talentoso empreendedor<sup>25</sup> apaixonado por análise de dados. Os três juntaram esforços para resolver um dos maiores problemas na área de software de análise de dados – *«tornar os dados de forma compreensível para pessoas comuns»*. Esses foram os ingredientes fundamentais para a criação do Tableau, que a cada dia que passa traz inovações na área de exploração de dados.

Em 2020, é previsível que o mundo vá gerar cinquenta vezes mais o volume de dados visto em 2011. Esses dados carregam dentro deles oportunidades enormes para o progresso humano. Contudo, para transformar essas oportunidades em realidades, as pessoas precisam de ter o poder dos dados nas suas mãos. É fundamentalmente, por estas razões que surge o Tableau, enquanto solução privilegiada para explorar esses dados, e que tem alterado a forma como as pessoas usam os dados para solucionar os seus problemas, proporcionando rapidez, facilidade, beleza estética, utilidade e flexibilidade na análise de dados. (Tableau, s.d).

Esta aplicação de análise multidimensional apresenta uma interface simplificada, com o recurso arrastar e soltar, e permite que o utilizador faça análises e apresentações com facilidade, e com liberdade criativa, com diversos tipos de visualizações dos dados, descobrindo novas informações, e tendências de negócio.

Possui vastos recursos de modelação como sejam, Drill Down, Drill Up, visões sumarizadas, detalhando informações da organização de uma forma interativa. Podemos assim, afirmar sem sombra de dúvida que é uma boa solução para melhorar a situação atual de análise de dados no sistema SPARES, auxiliando os investigadores na tomada de decisões sobre relações e eventos na atuação dos comissários no Santo Ofício.

---

<sup>23</sup> **Christopher Stolte**, is a Co-Founder of Tableau Software, Inc. and has been its Technical Advisor since August 2016.

<sup>24</sup> **Patrick M. Hanrahan** (born 1954) is a computer graphics researcher, the Canon USA Professor of Computer Science and Electrical Engineering in the Computer Graphics Laboratory at Stanford University.

<sup>25</sup> **Christian Chabot**, co-founded Tableau Software, Inc. and has been its Chairman of the Board since 2003 and Director since January 2003.

O Tableau permite ainda que os utilizadores façam perguntas e obtenham as respostas com mais facilidade e em menor tempo possível.

Como uma ferramenta líder de visualização de dados, o Tableau tem muitas características desejáveis e únicas. A sua poderosa aplicação de descoberta de dados e exploração permite-lhe responder a questões em segundos. O utilizador pode usar a interface de arrastar e soltar do Tableau para visualizar todos os dados, explorar diferentes pontos de vista, e até mesmo combinar várias fontes de dados com facilidade. É um tipo de ferramenta que não precisa de qualquer script complexo. Qualquer pessoa, que entenda o problema do negócio, pode analisá-lo com uma visualização dos dados considerada relevante. Quando a análise estiver concluída, pode partilhar o trabalho com outros utilizadores (w3ii, s.d).

### 6.2.1. Características de Tableau

O Tableau fornece soluções para todos os tipos de organizações e ambientes de dados.

Abaixo elencam-se as características únicas que fazem distinguir o Tableau da concorrência, e que de acordo com Gartner Group, o fazem ter um domínio sobre o mercado. (w3ii, s.d)

- **Velocidade na análise de dados**, dado que Tableau não exige que o utilizador tenha um nível elevado de conhecimento de programação. Qualquer utilizador pode analisar os dados e obter as informações necessárias com elevada rapidez;
- **Simplicidade na instalação e configuração**, a versão desktop é a mais utilizada, sendo a sua instalação muito simples e que contém todos os recursos necessários para iniciar a análise de dados sem restrição nenhuma;
- **Visualização diversificada**, o utilizador ao explorar os dados pode utilizar diversas formas de visualizar os dados, como gráficos, mapas, tabelas etc. Raramente faz-se o uso do script, sendo tudo feito de modo simples através do arrastar e soltar;
- **Relacionamento entre diversas fontes de dados**, pois permite que o utilizador combine diversas fontes de dados, em tempo real ou não, efetuando relacionamentos entre diversas fontes de dados, numa mesma pesquisa;
- **Funciona em todos os tipos de dispositivos**, não tendo o utilizador necessidade de se preocupar em que ambiente vai executar o Tableau, porque ele não necessita de nenhum hardware ou software específico para funcionar;
- **Colaboração em tempo real**, o Tableau pode filtrar, classificar e discutir os dados em tempo real. O utilizador pode gravar e partilhar a sua análise de dados e permitir que outros utilizadores possam observar os dados no seu navegador web.

## 6.2.2. Instalação e Configuração de Tableau



### Tableau Desktop: download gratuito e facilidade de uso

**Figura 25 – (Link para download do Tableau) <sup>26</sup>**

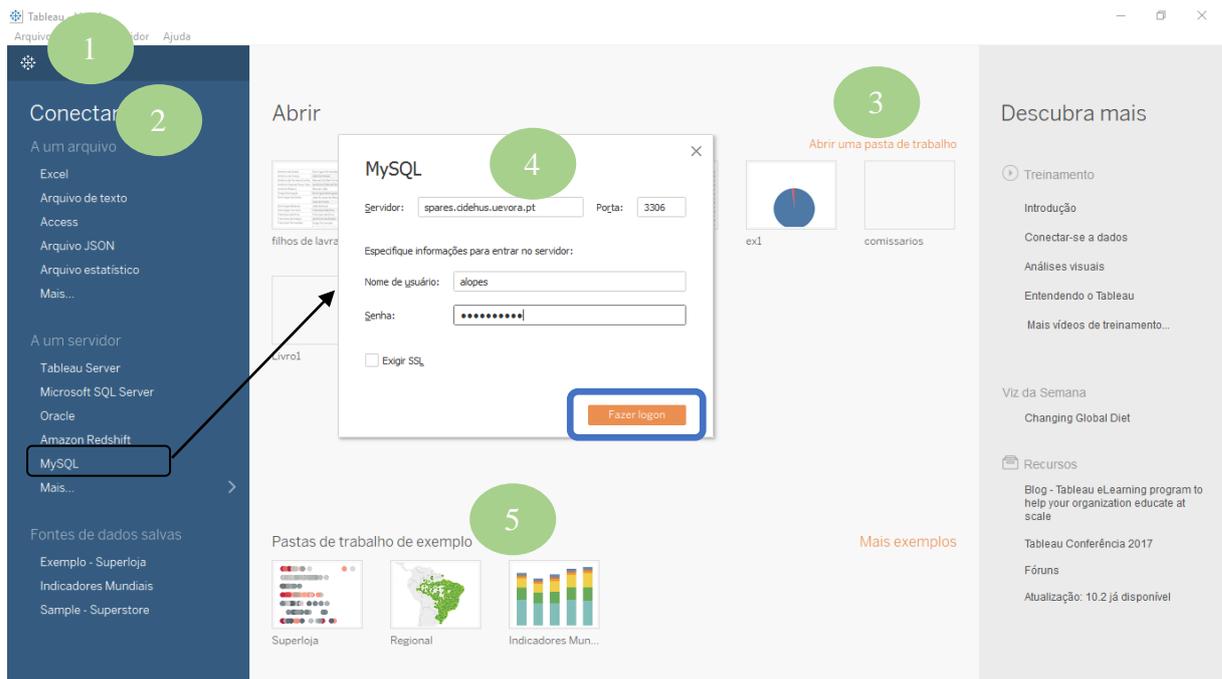
Para a análise de dados é necessário em primeiro lugar aceder ao link da Figura 25, e baixar a versão de Tableau para estudantes, professores e investigadores. A seguir, vai ser necessário preencher os dados pessoais do utilizador e enviar um documento válido que comprove a titularidade de aluno ou professor. De seguida, é avaliada a autenticidade do documento, e após a verificação é concedida a licença gratuita do Tableau, possibilitando-se dessa forma, a exploração de todas as suas funcionalidades. Para permitir a ligação entre sistemas de base de dados e o Tableau é necessário configurar os drivers ODBC que forem necessários.

### 6.2.3. Conectar aos dados do sistema SPARES ao Tableau

O Tableau oferece suporte à conexão com uma grande variedade de dados, armazenados em diversos lugares. Por exemplo, os dados podem estar armazenados no computador, em um ficheiro Excel, um arquivo de texto, um ficheiro Access, numa fonte de base de dados na nuvem, como o Amazon Redshift ou Salesforce, numa base de dados relacional, num Big Data, ou no servidor da organização, Oracle, SQL-Server, MySQL, entre outros.

Antes de criar uma exibição e analisar os dados, é indispensável a fase de conectar o Tableau aos dados do sistema SPARES. É necessário conectar o Tableau ao servidor «spares.cidehus.uevora.pt» onde encontra hospedada a base de dados SPARES, que funciona sobre um servidor MySQL.

<sup>26</sup> **Link para Download** - <https://www.tableau.com/pt-br/academic/students>

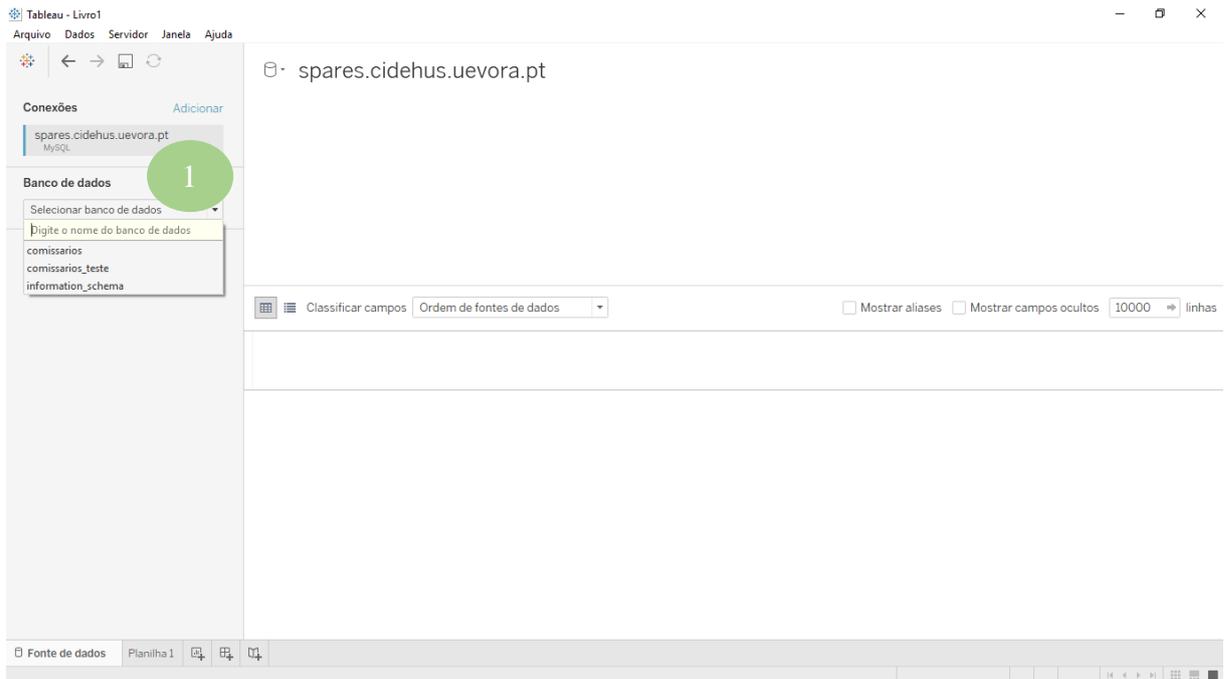


**Figura 26 – Passo 1 – Pagina inicial – Conectar ao servidor MySQL**

No quadrante esquerdo da página inicial encontram-se as listas das conexões possíveis. Em primeiro lugar, aparecem os tipos de arquivos, em seguida os tipos mais comuns de servidores, principalmente os utilizados recentemente. Para visualizar a lista completa das conexões possíveis do Tableau é necessário clicar no botão «*Mais...*». Consoante o tipo de conexão, serão solicitados diferentes formatos de informações para efetuar a ligação com o específico fonte de dados.

Essas são as opções relevantes facultadas na página inicial do Tableau conforme indicado na Figura 26:

1. O ícone  situado no canto superior esquerdo, permite em qualquer momento visitar a pagina inicial do Tableau.
2. Em conectar há possibilidade de:
  - a. De conectar a um arquivo;
  - b. Conectar a um servidor;
  - c. Conectar a fontes de dados que já tinham sido usadas recentemente;
3. Nesse ícone o utilizador pode abrir as pastas de trabalhos que tenham sido criadas;
4. Após escolher o servidor MySQL, essa janela permite introduzir as credenciais corretas para poder estabelecer a conexão com o sistema SPARES.
5. Exemplos de trabalhos realizados em Tableau.



**Figura 27 - Passo 2 - escolher a base de dados «comissarios»**

1. Após conectar ao servidor mysql apresenta-se uma lista de todas as bases de dados existente no servidor de acordo com a Figura 27. Para conectar ao sistema Spares deverá escolher-se a base de dados designada de «**comissarios**».

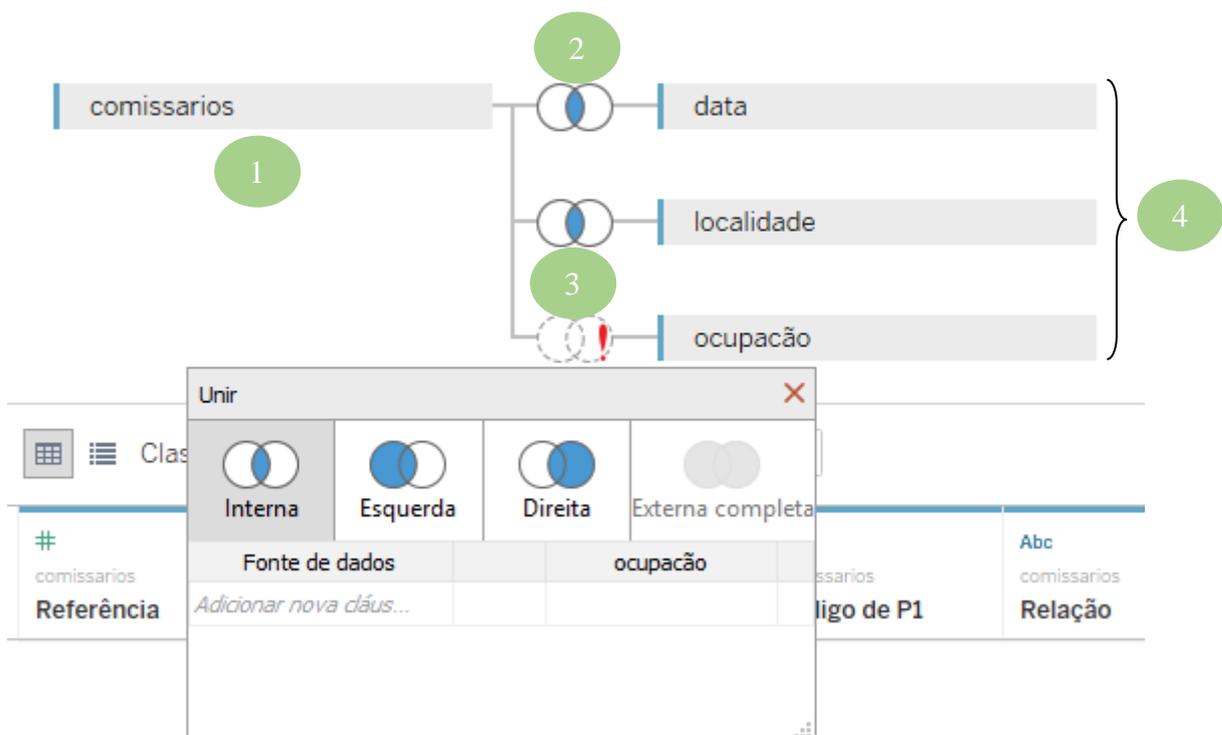


**Figura 28 - Lista de dimensões e tabela de factos do SPARES**

1. Na Figura 28 apresenta-se a lista das dimensões e tabela de facto do sistema SPARES.
2. Selecionar uma dimensão com o rato e arrastar para a área 2, mas somente as dimensões relacionadas com as análises de dados que queremos efetuar, sendo que os data mart

construídos para análise de dados no SPARES, encontram-se implementados segundo as recomendações do modelo dimensional, pelo que quando efetuamos a análise desses dados, geralmente, a tabela que arrastamos primeiro é a tabela de factos «comissários» que é a tabela central do esquema em estrela. De seguida arrastamos as dimensões, de acordo com a pesquisa que pretendemos efetuar, por exemplo, se estamos a pesquisar os eventos ocorridos numa determinada localidade, não temos a necessidade de adicionar outras dimensões além da dimensão «localidade».

## comissarios+ (comissarios\_teste)



**Figura 29 – Passo 3 - Estabelecer a ligação entre tabela de factos e as dimensões**

Podemos observar na Figura 29, as dimensões que estão ligadas à tabela de factos sendo, após ser arrastada para essa área, estabelecidas as ligações automaticamente, entre a chave primária da dimensão, que é chave estrangeira na tabela de factos.

1. O primeiro a ser adicionado à área é a tabela de facto «comissários»;
2. Constatamos que a dimensão data, como tem ligação com a tabela de factos, faz com que o Tableau reconheça essa união como interna; em comando SQL seria o INNER JOIN.
3. Esse é um exemplo prático onde a dimensão ocupação, apesar de ter um atributo correspondente no «comissários», não contém nenhuma chave estrangeira na tabela de factos. Deste modo, o Tableau não reconhece nenhuma ligação entre ocupação e comissários, por isso observamos um ponte de exclamação vermelho.
4. As tabelas de dimensões adicionadas para análise de dados.

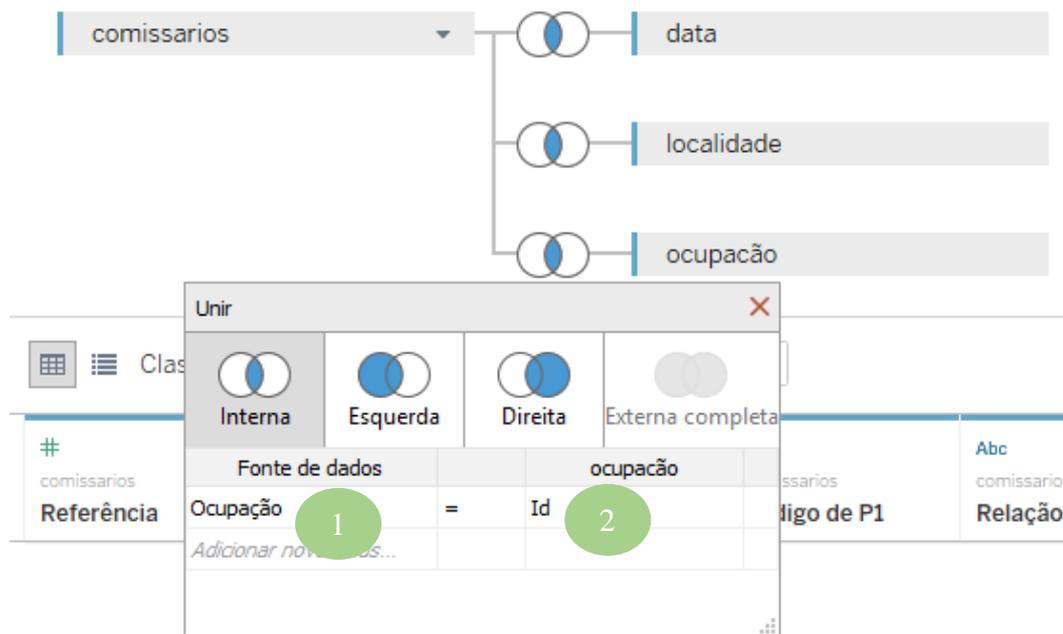


Figura 30 - Passo 4 - Estabelecer a ligação entre tabela de factos e as dimensões

A Figura 30 demonstra a resolução do problema apresentado no ponto 3 da Figura 29. Basicamente, o utilizador deve conhecer o sistema, para saber qual é o campo que tem relação entre as tabelas. Por isso, é aconselhável que as análises sejam feitas com as relações do esquema em estrela, de modo a não ocorrerem erros nas ligações. É importante perceber que esse caso é um exemplo, porque na realidade a ocupação será uma chave estrangeira na tabela «comissarios».

1. Coluna «ocupação» da tabela de facto.
2. Coluna Id na dimensão «ocupação» é chave primária.

#	comissarios	comissarios	comissarios	#	comissarios	comissarios	#	comissarios	comissarios
Referência	Nome de P1	Sigla de P1	Código de P1	Relação	Sigla de P2	Código de P2	Nome de P2	Data	
82	Manuel de Faria e Abr...	P	164	Testemunha na habili...	P	178	Manuel de Faria e Al...	1	
86	Manuel de Faria e Abr...	P	164	Testemunha na habili...	P	179	António Vogado Sout...	1	
364	nulo	nulo	nulo	nulo	P	267	Bernardo Bruloughan...	1	
365	nulo	nulo	nulo	nulo	P	267	Bernardo Bruloughan...	1	
368	nulo	nulo	nulo	nulo	P	268	Tomás Machugo, Pad...	1	
369	nulo	nulo	nulo	nulo	P	268	Tomás Machugo, Pad...	1	
1419	António José Pereira, ...	P	486	Escrivão ad hoc na ha...	P	523	Nicolau José Antunes...	1	
1448	António José Pereira, ...	P	486	Voto favorável no Co...	P	95	Luís Barata de Lima	1	
1574	Manuel Álvares da Co...	P	530	Local preciso da habil...	P	1	Hugo Maguiere	1	

Figura 31 - Preparação do dados em tempo real

No ponto 1 da Figura 31 podemos visualizar as duas formas de análise de dados, em tempo real e na memória. Normalmente, utilizamos a exploração de dados em tempo real. A figura demonstra-nos que já contém todas as configurações necessárias para iniciar a analisar os dados.

### 6.2.4. Ambiente de trabalho do Tableau

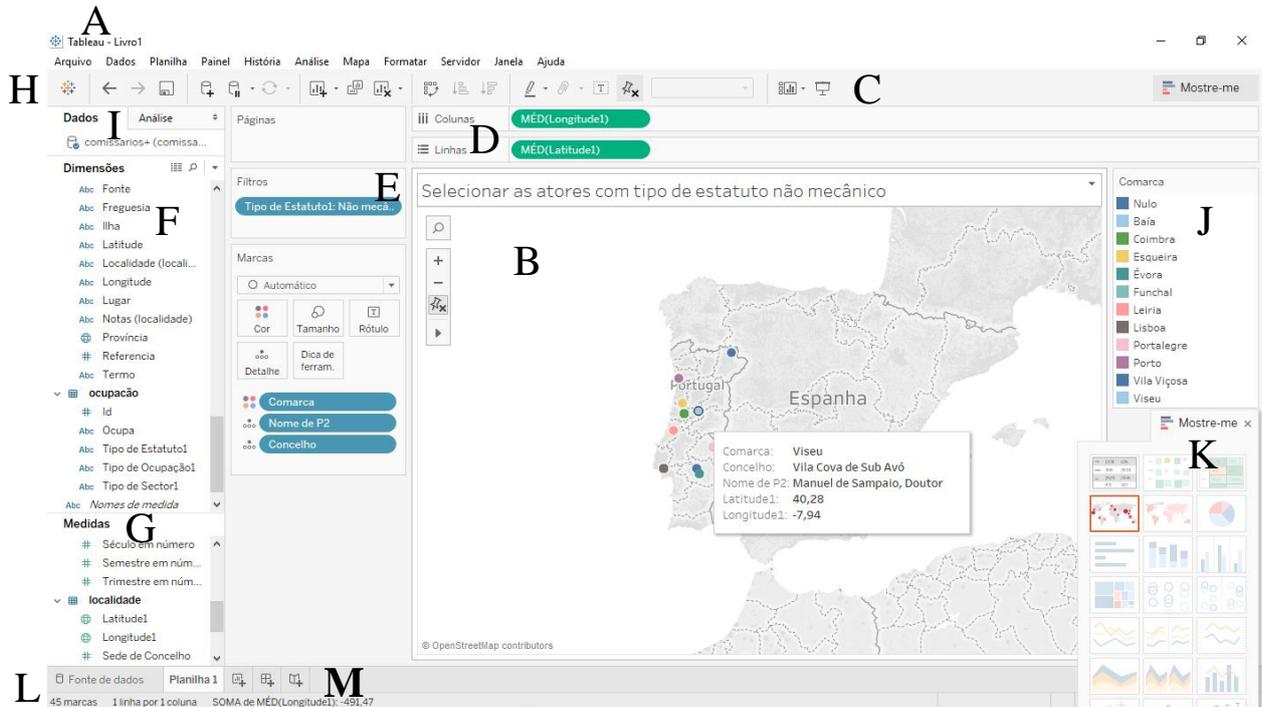


Figura 32 - Ambiente de trabalho de Tableau

- a. Nome da pasta de trabalho.
- b. Exibição - Este é o ambiente de trabalho, onde o utilizador pode observar os seus outputs em diferentes tipos de visualizações.
- c. Barra de ferramentas - a barra de ferramentas serve para os comandos e ferramentas de análise e navegação.
- d. Área para adicionar linhas e colunas a serem analisadas. Somente com um clique arrastar as medidas ou dimensões para serem analisadas.
- e. Área de adicionar os filtros.
- f. Lista das dimensões.
- g. Lista das medidas.
- h. Ir para página inicial.
- i. Adicionar nova fonte de dados, e editar relacionamento entre as fontes.
- j. Barra lateral a representar os filtros por diferentes cores.
- k. Barra lateral, com os diferentes tipos de visualizações de dados.

- l. Retroceder para a página da fonte de dados.
- m. Barra horizontal com os ícones para criar uma nova exibição, ou novo painel ou nova história.

### 6.2.5. Tipos de dados em Tableau

Quadro 21 - Tipos de dados em Tableau. Adaptado de: (Tableau, s.d)

<i>Ícone</i>	<i>Tipo de dados</i>
Abc	Valores de textos
📅	Valores da data
🕒	Valores de data e hora
#	Valores numéricos
T F	Valores booleanos
🌐	Valores geográficos (usados com mapas)

Todos os atributos em uma fonte de dados têm determinado tipo de dados. O tipo de dados condiz com o tipo de informação armazenada nesse campo; A título de exemplo, os tipos de dados inteiros (10), as datas (10/06/1991), e textos que são sequências de caracteres (“Fogo”). O Quadro 21 apresenta os ícones correspondentes a cada tipo de dado, como são representados no Tableau.

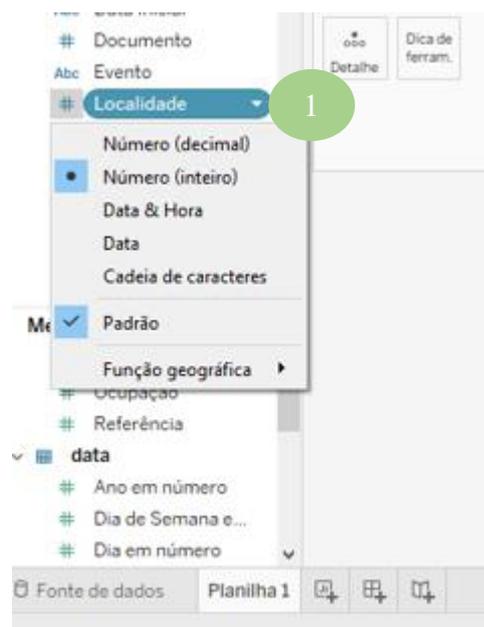


Figura 33 - Alterar tipo de dados de um atributo

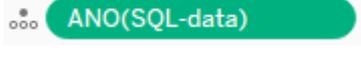
Por vezes, o Tableau pode interpretar erroneamente um determinado tipo de dados de um atributo. Por exemplo, o Tableau pode considerar um atributo que contém datas como um tipo de dados inteiro, e não como um tipo de dados de data.

**Nota:** é sempre necessário verificar antes de efetuar a análise de dados, se o tipo de dados do atributo a ser analisado está de acordo com o seu tipo na fonte de dados. Caso não esteja, há a possibilidade de modificar o tipo de dados de um atributo; um exemplo clássico é com os códigos postais, pois apesar dos valores serem números, não são contínuos, devido ao facto que não poderem ter somados ou fazer médias com os códigos postais. No caso do Tableau adicionar à área de medidas, ela deve ser arrastada para a área de dimensões.

1. Clique no ícone de tipo de campo, conforme mostrado na Figura 33, e escolha o novo tipo de dados que preferir para esse atributo.

### 6.2.6. Dimensões e Medidas

Quadro 22 - Tipos de medidas e dimensões.

Ícone	Medidas e Dimensões
	Dimensão Discreta
	Medida Contínua
	Medida Discreta
	Dimensão Contínua

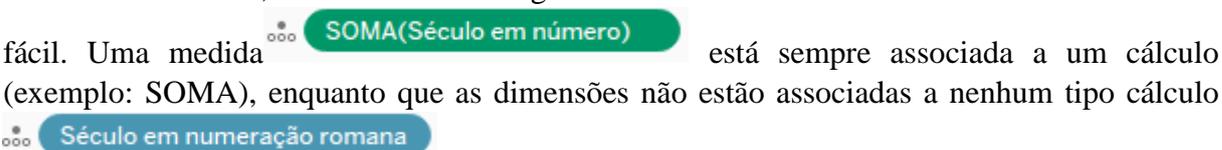
No momento em que o utilizador efetuar as configurações bem definidas e escolher as dimensões para análise de dados, automaticamente, o Tableau atribui a cada atributo na fonte de dados como sendo uma «dimensão» ou uma «medida».

De uma forma geral, as dimensões são discretas e as medidas são contínuas, mas na prática, todas as combinações são possíveis, e uma dimensão pode ser contínua assim como uma medida pode ser discreta. Nos campos com valores de textos e booleanos é adicionada a área de dimensões, e nos campos numéricos e dados georreferências é colocada a área de medidas.

Inicialmente, qualquer dimensão ao ser adicionada na exibição será discreta com o fundo azul. De início, é intuitivo distinguir uma dimensão de uma medida através das cores. Um erro muito frequente, é assumir que a cor azul representa as dimensões, e a cor verde as medidas, quando na verdade é a cor azul que representa os dados discretos e a verde os dados contínuos, como aliás se pode comprovar no Quadro 22.

O Tableau mostra um eixo quando o utilizador arrastar e solta um campo contínuo em linhas ou colunas na área de exibição. Analogicamente, o eixo representa uma régua que demonstra os valores entre mínimos e máximos daquele atributo, enquanto que, se for um campo discreto a ser adicionado na linha ou coluna da exibição ele cria um cabeçalho.

É importante salientar, que o Tableau não faz agregação das dimensões. No caso de o utilizador pretender que um determinado atributo da dimensão seja agregado, necessitará de promover a conversão desse atributo para ser uma medida. Agregação significa basicamente, selecionar um conjunto de valores, por exemplo saber o número de eventos ocorridos num determinado intervalo de tempo, sendo então necessário converter o atributo «Evento» de discreto para contínuo para ser possível efetuar a sua contagem.

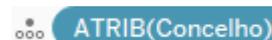
Visualmente, a técnica de distinguir entre uma dimensão e uma medida é bastante fácil. Uma medida  está sempre associada a um cálculo (exemplo: SOMA), enquanto que as dimensões não estão associadas a nenhum tipo cálculo

### 6.2.7. Agregações em Tableau

Pontualmente, é importante examinar os dados numéricos em formato agregado, como um somatório ou uma média. As funções matemáticas que produzem dados agregados são chamadas de funções de agregação. As funções de agregação efetuam um cálculo num conjunto de valores que resultam em um único valor. Por exemplo, uma medida que contenha os valores 5, 9, 3, 4 agregados como uma soma retornará um valor único: 21. Ou, caso uma farmácia realize 3.000 transações diárias de vendas de 100 medicamentos diferentes, se o utilizador preferir exibir a soma das vendas de cada medicamento, para que possa decidir quais medicamentos teve a receita mais alta (Tableau, s.d).

O Tableau fornece um conjunto de agregações predefinidas que são apresentadas em linhas seguidas, tendo em conta que a medida contém os seguintes valores [1, 2, 2, 3]:

- **Atributo**, retornará o valor da expressão especificada somente se ela tiver um único valor para todas as linhas no grupo; caso contrário exibirá um caractere de asterisco (\*). Esta agregação é particularmente útil ao agregar uma dimensão. Para definir uma medida na exibição como essa agregação, clique com o botão direito do rato na medida e escolha **Atributo**. Não tem nenhum resultado. O campo mudará para mostrar o texto ATRIB:



- **Contagem**, retorna o número de linhas em uma medida ou dimensão. Quando aplicado a uma dimensão, o Tableau cria uma nova coluna temporária que é uma medida porque o resultado de uma contagem é um número. Podem ser contados números, datas, operadores booleanos e textos. Os valores nulos são ignorados em todos os casos. Para contar uma medida ou uma dimensão, clique com o botão

direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Contagem**. Com os valores definidos o resultado seria o valor (4). O campo mudará para mostrar a função CONT:

☰ CONT(Nome de P2)

- **Contagem (distinta)**, esta agregação é idêntica à função contagem, sendo que a única diferença é que a contagem é distinta, não englobando os valores repetidos. Nesses valores [1, 2, 2, 3] a contagem retorna 4 elementos, mas a contagem distinta retorna somente 3, porque não conta o número dois duas vezes. Clique com o botão direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Contagem (distinta)**. Com os valores definidos o resultado seria o valor (3). O campo mudará para mostrar a função CONTD:

☰ CONTD(Nome de P2)

- **Desagregar**, retorna todos os registos na fonte de dados subjacente. Para desagregar todas as medidas na exibição, selecione **Agregar medidas** no menu Análise (para remover a marca de seleção). Quando os dados são desagregados, é possível mostrar todas as linhas individuais da fonte de dados. Volta tudo como no início, sem nenhuma agregação nos dados. O resultado será quatro valores (1, 2, 2, 3).
- **Dimensão**, retorna todos os valores exclusivos em uma medida ou dimensão. clique com o botão direito do rato na medida ou dimensão e escolha Dimensão. o resultado é três valores (1, 2, 3).
- **Máximo**, retorna o maior número em uma medida ou dimensão contínua. Os valores nulos são ignorados. Para conhecer o valor máximo de medida ou uma dimensão contínua, clique com o botão direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Máximo**. O resultado é o valor (3). O campo mudará para mostrar a função MAX:

☰ MÁX(Século em número)

- **Médias**, retorna a média aritmética dos números em uma medida. Os valores nulos são ignorados. Clique com o botão direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Média**. O resultado é o valor (2). O campo mudará para mostrar a função MED:

☰ MÉD(Número de registros)

- **Mediano**, retorna o mediano dos números em uma medida. Os valores nulos são ignorados. Ela não está disponível para conexões usando qualquer uma das fontes de dados a seguir:
  - a) Access
  - b) Amazon Redshift
  - c) Cloudera Hadoop

- d) HP Vertica
- e) IBM DB2
- f) IBM Netezza
- g) Microsoft SQL Server
- h) MySQL**
- i) Teradata

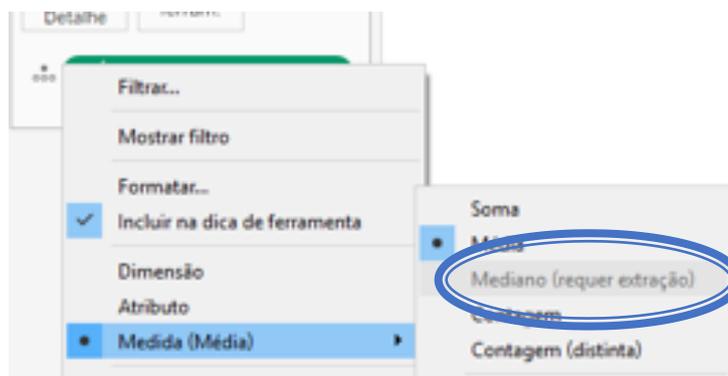


Figura 34 - Agregação Mediano em Tableau

A Figura 34 demonstra que como estamos a usar MySQL, esta função não se encontra disponível. O resultado será o valor (2).

- **Mínimo**, retorna o menor número em uma medida ou dimensão contínua. Os valores nulos são ignorados. Para conhecer o valor mínimo de medida ou uma dimensão contínua, clique com o botão direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Mínimo**. O resultado é o valor (1). O campo mudará para mostrar a função MIN:

MIN(Trimestre em número)

- **Desvio Padrão**, retorna o desvio padrão de todos os valores na expressão indicada com base em uma população de amostra. Os valores nulos são ignorados. Retornará um valor Nulo se houver menos que dois membros no exemplo que não sejam nulos. Use esta função caso os dados representem uma amostra da população. O resultado é o valor (0,8165).

DESVPAD(Código de P2)

- **Desvio Padrão Pop**, retorna o desvio padrão de todos os valores na expressão indicada com base em uma população de viés. Presume que os argumentos consistem na população inteira. Use esta função para tamanhos de amostra maiores. clique com o botão direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Desvio Padrão (pop)**. O campo mudará para mostrar a função DESVPADP:

DESVPADP(Trimestre em número)

- **Percentil**, retorna o valor no percentil especificado para a medida. Ao selecionar esta agregação, você deve escolher em um submenu com diversos valores de percentil: 5, 10, 25, 50, 75, 90, 95. Resultado é um valor. O valor de PCT50 seria 2 para os dados indicados.



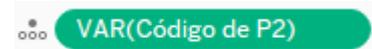
Figura 35 - Agregação Percentil em Tableau

Não se encontra disponível para MySQL como podemos observar na Figura 35.

- **Soma**, devolve a soma dos números em uma medida. Os valores nulos são ignorados. Para conhecer o valor da soma de medidas ou uma dimensão contínua, clique com o botão direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Soma**. O resultado baseado no intervalo definido é o valor (8). O campo mudará para mostrar a função SOMA:



- **Variância**, retorna a variação de todos os valores na expressão indicada com base em uma amostra. Os valores nulos são ignorados. Retornará um valor Nulo se houver menos que dois membros no exemplo que não sejam nulos. Use esta função caso os dados representem uma amostra da população. Clique com o botão direito do rato na medida ou dimensão e escolha **Medida** e depois escolha **Variância**. O campo mudará para mostrar a função VAR. O resultado será o valor (0,6667).



- **Variância (pop)**, retorna a variação de todos os valores na expressão indicada com base em uma população de viés. Presume que os argumentos consistem na população inteira. Use esta função para tamanhos de amostra maiores. O resultado é um valor (0,5000).



Existe também a possibilidade de definir agregações personalizadas. Dependendo do tipo de exibição de dados criado pelo utilizador, o Tableau aplicará essas agregações no nível de detalhe apropriado. Por exemplo, o Tableau aplicará a agregação de soma de números de registos obtidos no momento que cria uma exibição.

### 6.2.8. Navegação em Tableau

Este capítulo serve de auxílio ao utilizador, para que se familiarize com os vários recursos de navegação disponíveis na interface do Tableau. Temos vários menus na parte superior, com os respetivos comandos de navegação. Infra indicam-se alguns detalhes relevantes de cada menu e as suas características principais (w3ii, s.d):

- **Menu Arquivo**, este menu é utilizado para criar um novo livro Tableau e abrir pastas de trabalho existentes, tanto de sistema local como servidor Tableau. As características importantes neste menu são:
  - a) Localidade da pasta de trabalho, para definir o idioma a ser utilizado no relatório;
  - b) Local de repositório, para colar uma folha na pasta de trabalho atual, que é copiado de outro livro;
  - c) Exportar pasta de trabalho em pacote, é usado para criar uma pasta de trabalho que será compartilhado com outros usuários.
  
- **Menu dados**, este menu é utilizado para criar uma nova fonte de dados para buscar os dados para análise e visualização. De igual modo, permite que o utilizador substitua ou atualize uma fonte de dados existente. As características cruciais deste menu são as seguintes:
  - a) Nova fonte de dados, para ver todos os tipos de conexões disponíveis e escolher a que o utilizar precisa;
  - b) Atualizar todas as extrações, para atualizar a fonte de dados;
  - c) Editar relacionamentos, sendo usado para definir os campos relacionáveis entre duas fontes de dados, efetuando ligação entre eles.
  
- **Menu folha de cálculo (Planilha)**, este menu é utilizado para criar uma nova folha de cálculo, juntamente com vários recursos de exibição, como mostra o título e subtítulos, entre outros. As características principais deste menu são as seguintes:
  - a) Mostrar resumo, para ver o resumo dos dados utilizados na folha de cálculo, a título de exemplo temos a função contar;
  - b) Dica de ferramenta, para mostrar a dica de ferramenta sobre vários campos de dados;
  - c) Atualizações automática, é usado para atualizar os dados da folha de cálculo ou filtros usados.
  
- **Menu painel**, este menu é utilizado para criar um novo painel de instrumentos, juntamente com vários recursos de exibição, ou seja, criar dashboards, que pode ser apresentado em diversos tipos de formatos. As características primordiais neste menu são as seguintes:
  - a) Formatar, é utilizado para definir o layout em termos de cores e seções do painel;
  - b) Ações, para ligar as folhas do painel para URLs externos ou outras folhas;

- c) Exportar imagem, é usado para exportar uma imagem do Dashboards.
- **Menu de história**, este menu é usado para criar uma nova história que tem muitas folhas ou painéis com dados relacionados. As características importantes neste menu são as seguintes:
  - a) Formatar é usado para definir o layout em termos de cores e seções da história;
  - b) Executar atualização, para atualizar a história com a mais recente fonte de formulário de dados;
  - c) Exportar imagem, é usado para exportar uma imagem da história.
- **Menu de análise**, este menu é utilizado para analisar os dados presentes na folha. O Tableau fornece muitos recursos, tais como calcular a percentagem e fazer uma previsão etc. As características importantes neste menu são as seguintes:
  - a) Linhas de tendências, para mostrar a linha de tendência para séries de dados;
  - b) Criar campo calculado e editar campo calculado, é usado para criar campos adicionais com base em determinado cálculo em campos existentes, e editar esses campos criados de acordo com a pesquisa do utilizador;
  - c) Agregar medidas, tem utilidade se o utilizador pretender desagregar todas as medidas a serem analisadas.
- **Menu mapa**, este menu é utilizado para a construção de visualizações de mapas em Tableau. O utilizador pode atribuir funções geográficas para campos nos seus dados. As suas principais características são as seguintes:
  - a) Mapas em segundo plano, para ocultar e mostrar as camadas do mapa, tais como nomes de ruas e fronteiras do país, e adicionar camadas de dados;
  - b) Codificação geográfica, para criar novos papéis geográficos e atribuí-los aos campos geográficos nos seus dados.

### 6.2.9. Fluxo de projeto em Tableau

A Figura 36 apresenta-nos o ciclo de vida de um projeto em Tableau. É de vital importância, dado que para analisar os dados é necessário um planeamento muito meticuloso, para criar dashboards úteis e atrativos, e criar uma boa história.

Normalmente, o resultado final de um projeto em Tableau consiste em criar os painéis, ou seja, apresentar os dados em dashboards e scorecards. Mas para isso há diversos passos intermediários que precisam de ser efetuados com precisão a fim de atingir os objetivos pretendidos. A Figura 36 apresenta um fluxo de etapas do projeto que deve ser idealmente seguido para criar dashboards eficazes no Tableau.

Resumidamente, o utilizador necessita em primeiro de tudo conectar a fonte de dados pretendido para análise, criar e editar a vistas de dados para isso Tableau fornece recursos de arrastar e soltar facilitando o modo de análise. Em seguida as vistas criadas precisam ser reforçadas pelo uso de filtros, agregações entre outras formatações. posteriormente ao criar

uma folha de cálculo, o utilizador deve de criar diferentes pontos de vista sobre os mesmos dados ou dados diferentes, finalmente criar painéis e história contendo várias folhas de cálculos que estão ligadas entre si, apresentado em forma de dashboards.

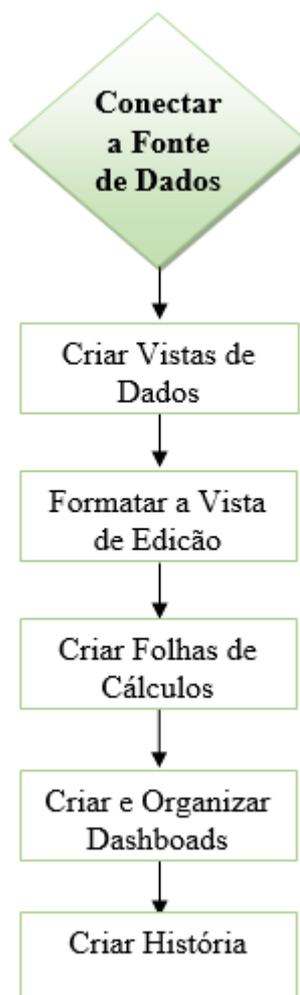


Figura 36 - Fluxo do projeto em Tableau

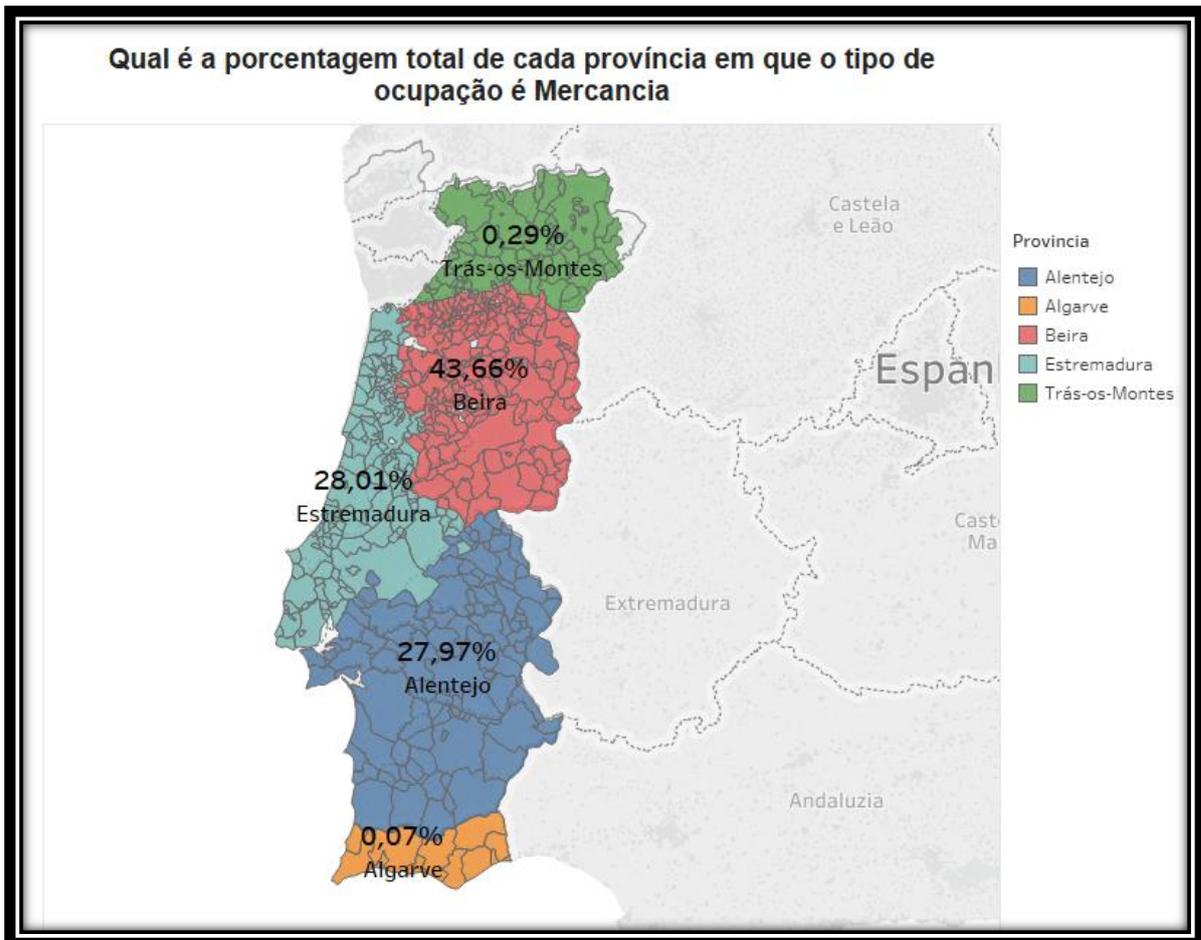
### 6.2.10. Georreferenciação no Sistema SPARES

Observando a Figura 37 constatamos que o sistema de georreferenciação, não tem uma boa qualidade de dados nem dados confiáveis, devido à alta desintegridade dos dados. A dimensão localidade tem a integridade referencial, mas não tem a integridade dos dados, o que nos leva a identificar alguns problemas com a georreferenciação, destacando-se em particular três problemas.

O primeiro problema foi detetado pelo Tableau, que consiste nas coordenadas geográficas. No momento em que executamos uma pesquisa no Tableau com as coordenadas geográficas, o Tableau não conseguia reconhecer como coordenadas geográficas. Constatamos que o campo de latitude e longitude era de tipo texto «varchar». Para resolver o problema, criei um procedimento que se encontra no Anexo E, que adiciona um novo campo

de latitude e longitude com o tipo de dados «Float Point» à dimensão localidade, e extraí os dados dos campos antigos, convertendo de texto para números flutuantes, e carreguei nas novas colunas criadas.

Ao finalizar, eliminei a coluna antiga, e o problema ficou resolvido. A partir desse momento, ao efetuarem-se pesquisas em Tableau com as coordenadas geográficas, o aplicativo tipo cubo reconhece de imediato que se trata de campo com coordenadas geográficas e imprime os resultados em mapas.

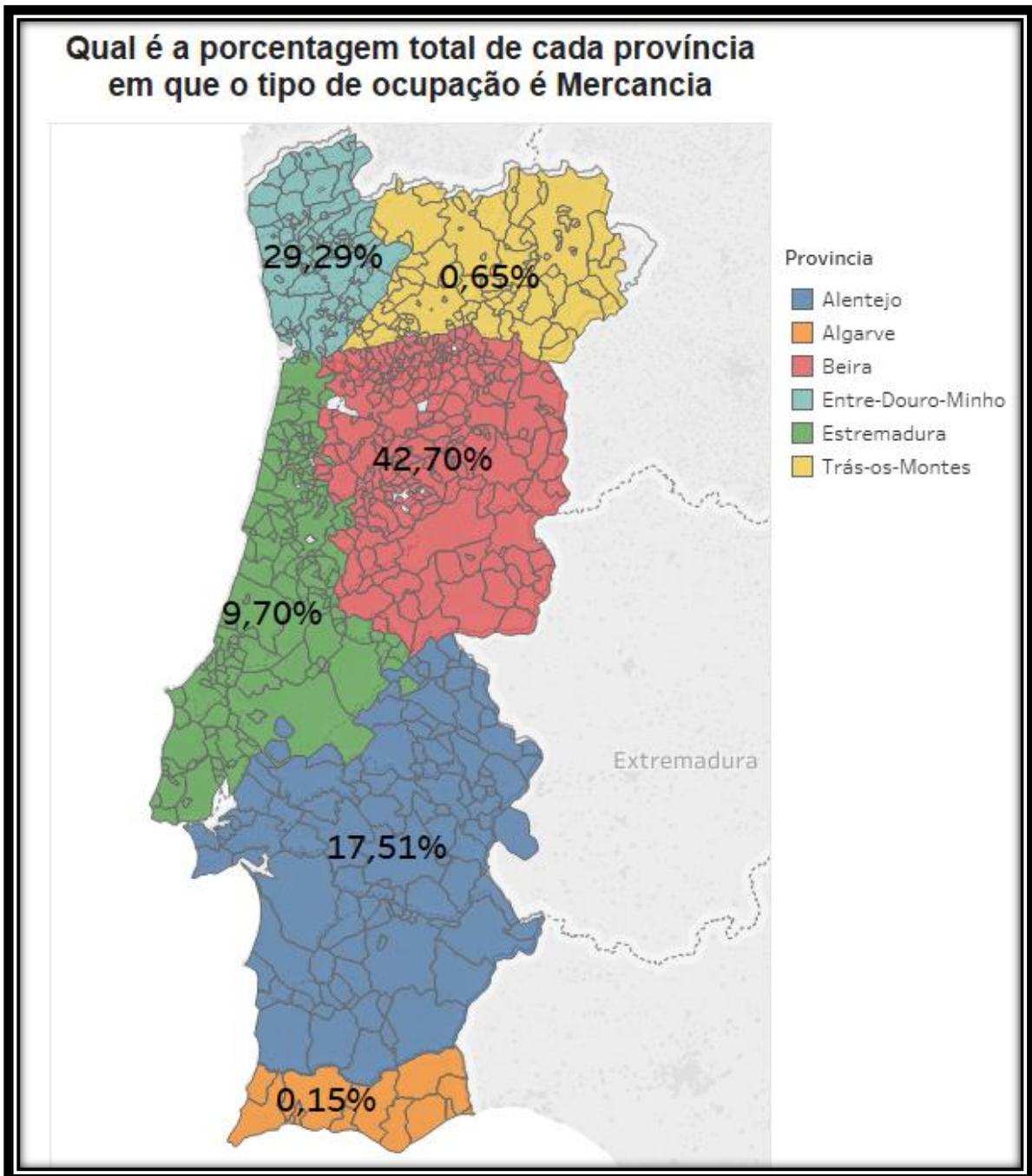


**Figura 37 – Análise de percentagem total do tipo ocupação Mercancia em cada província com a Georreferenciação no SPARES actual**

O segundo problema, também foi detetado pelo Tableau, ao realizar a análise consoante a Figura 37 demonstra. Se observamos bem a figura, há uma província onde não aparecem os dados, que é a província de «Entre Minho e Douro». A razão associado a esse facto, é que não havia integridade de dados, que é a manutenção e a garantia da precisão e consistência de dados durante todo o ciclo de vida da informação.

A província do Minho era escrita de diversas formas, tendo a forma única de escrever surgido, ao tentar resolver o terceiro problema, que basicamente era imprimir os resultados de acordo com o mapa de Portugal do ano 1527.

O mapa das províncias de Portugal da Idade Moderna tem algumas diferenças com o mapa atual. A título de exemplo podemos registar que Olivença hoje pertence a Espanha, mas até 1801 pertencia à província do Alentejo. Os investigadores e utilizadores do sistema SPARES, têm a necessidade de cartografar os resultados tanto em microescala como em macro escalas, de acordo com o mapa da Idade Moderna.



**Figura 38 - Análise de percentagem total do tipo ocupação Mercancia em cada província com a Georreferenciação no novo SPARES**

A versão 10.3 do Tableau já contém uma fonte de dados para arquivos espaciais, o que possibilita a análise de dados com mapas no formato shapefile. Dado que o CIDEHUS possui um mapa de Portugal com todos os concelhos e as províncias da Idade Moderna, podemos

ligar o Tableau com o arquivo espacial e a posteriori ligamos o arquivo shapefile à dimensão de localidade, onde o campo «concelho» liga com o campo «Nome Conc» no arquivo espacial, ou podemos efetuar a ligação através da província.

Após a ligação estabelecida de forma correta, efetuei a pesquisa referente à Figura 37, tendo constatado que a província Entre Minho e Douro não apresentava os resultados. O problema foi identificado, chegando à conclusão que no arquivo shapefile aparecia escrito «Entre-Minho-Douro» e no sistema SPARES estava escrito «Entre Minho e Douro». Para resolver o problema, criei um procedimento que se encontra no anexo D, e que resumidamente estabelece a alteração da escrita da província «Entre Minho e Douro» para «Entre-Minho-Douro» no sistema SPARES.

Podemos notar na Figura 38, que depois do tratamento dos dados, já temos informações com boa qualidade, e já temos o resultado na província de Minho. Chegamos à conclusão que a província de Beira, Entre-Minho-Douro e Alentejo tem maior número de percentagem de pessoas ligadas ao tipo de ocupação Mercancia.

### 6.2.11. Filtros em Tableau

Anteriormente o filtro era a única ferramenta de pesquisa de dados no SPARES. Com a aplicação de técnicas e Business intelligence, os filtros tornaram-se uma ferramenta utilizada no Tableau. O processo de filtragem, é o processo de remoção de certos valores ou gama de valores a partir de um conjunto de resultados. No Tableau existem três tipos básicos de filtros a mencionar:

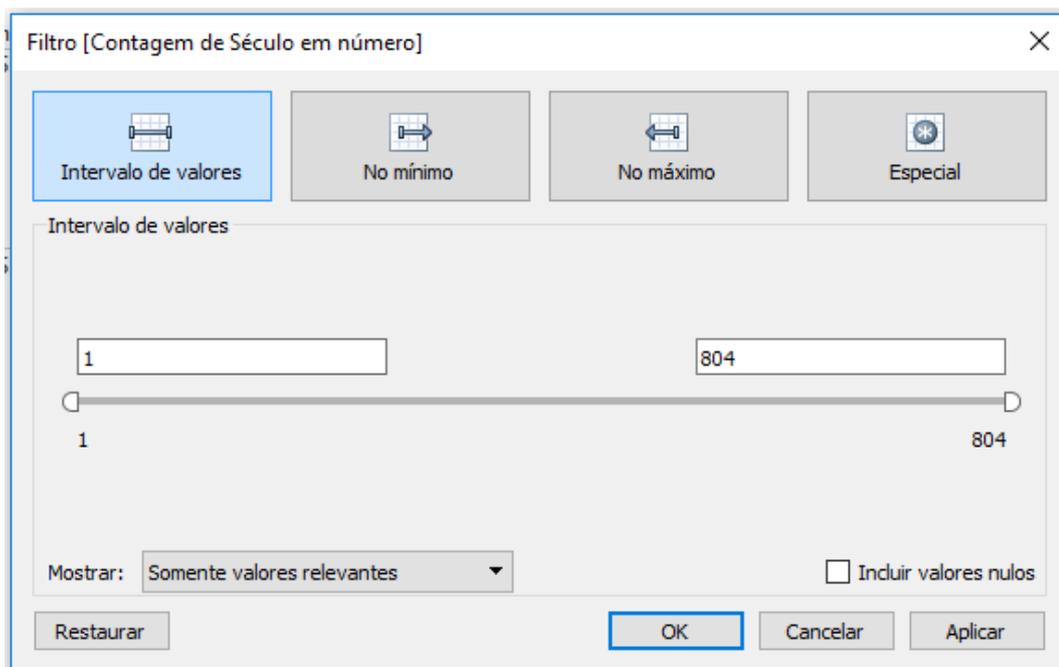


Figura 39 - Filtros em medidas

- **Filtro de medidas**, aplicado nos campos de medida, é baseado nos cálculos aplicados aos campos de medida. Assim, enquanto nos filtros de dimensão usamos apenas valores para filtrar, no filtro de medidas usamos cálculos baseados em campos calculados. Se observamos a Figura 39, verificamos que podemos efetuar filtros num intervalo de valores, ou parametrizar por valores mínimos ou máximos, e ainda há a opção de criar um filtro especial sobre os valores com código SQL.

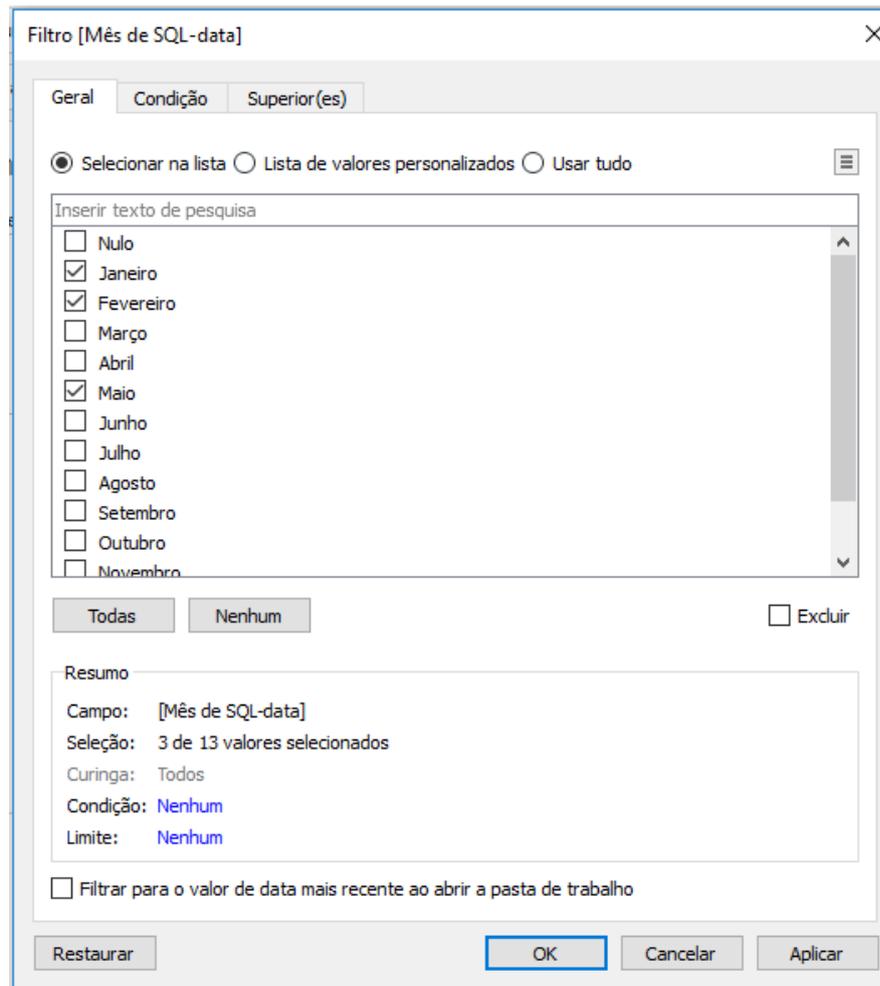
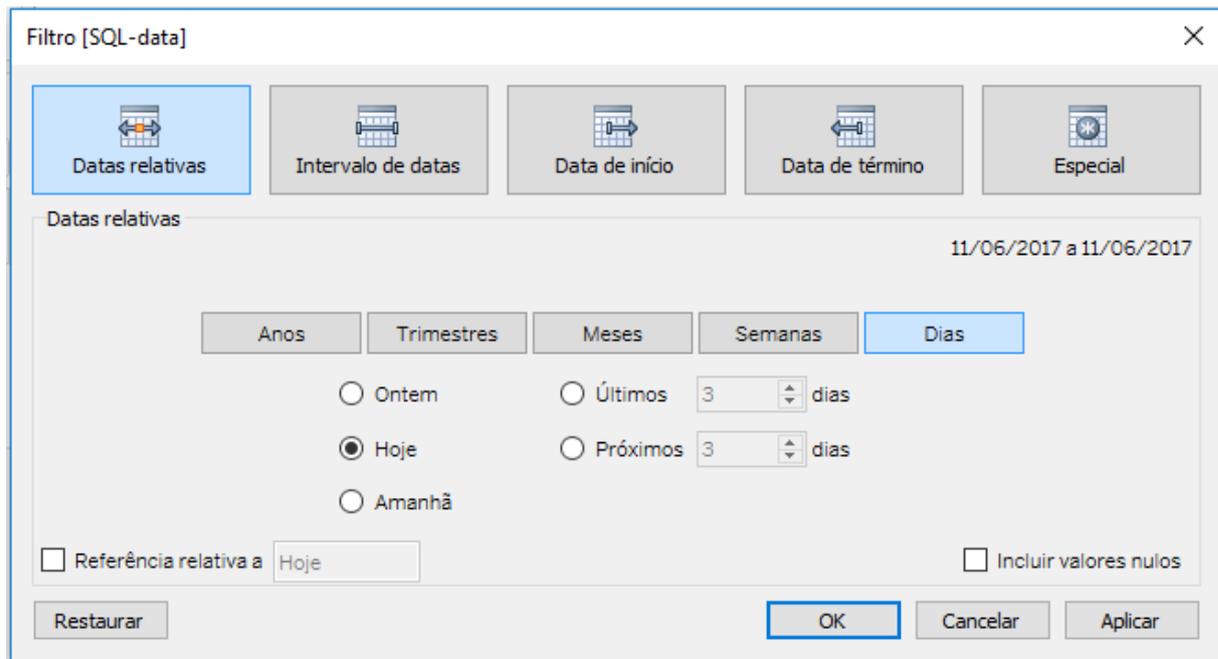


Figura 40 - Filtros em dimensões

- **Filtro de dimensões**, são aplicados nos campos de dimensão, a título de exemplo, podemos referir filtragem de texto, valores numéricos ou temporais, usando as expressões lógicas e condicionais. Como podemos observar na Figura 40, aplicamos um filtro na dimensão «meses», em que aparece a lista do mês, e temos a opção de selecionar todos, ou selecionar os personalizados. No caso como eram muitos valores, temos a opção de escrever a palavra que pretendemos e ela apresenta somente os valores que contêm a sequência do texto escrita. Ainda temos a opção de efetuar filtros por condições por campo ou por fórmulas ou script SQL.



**Figura 41 - Filtros em datas**

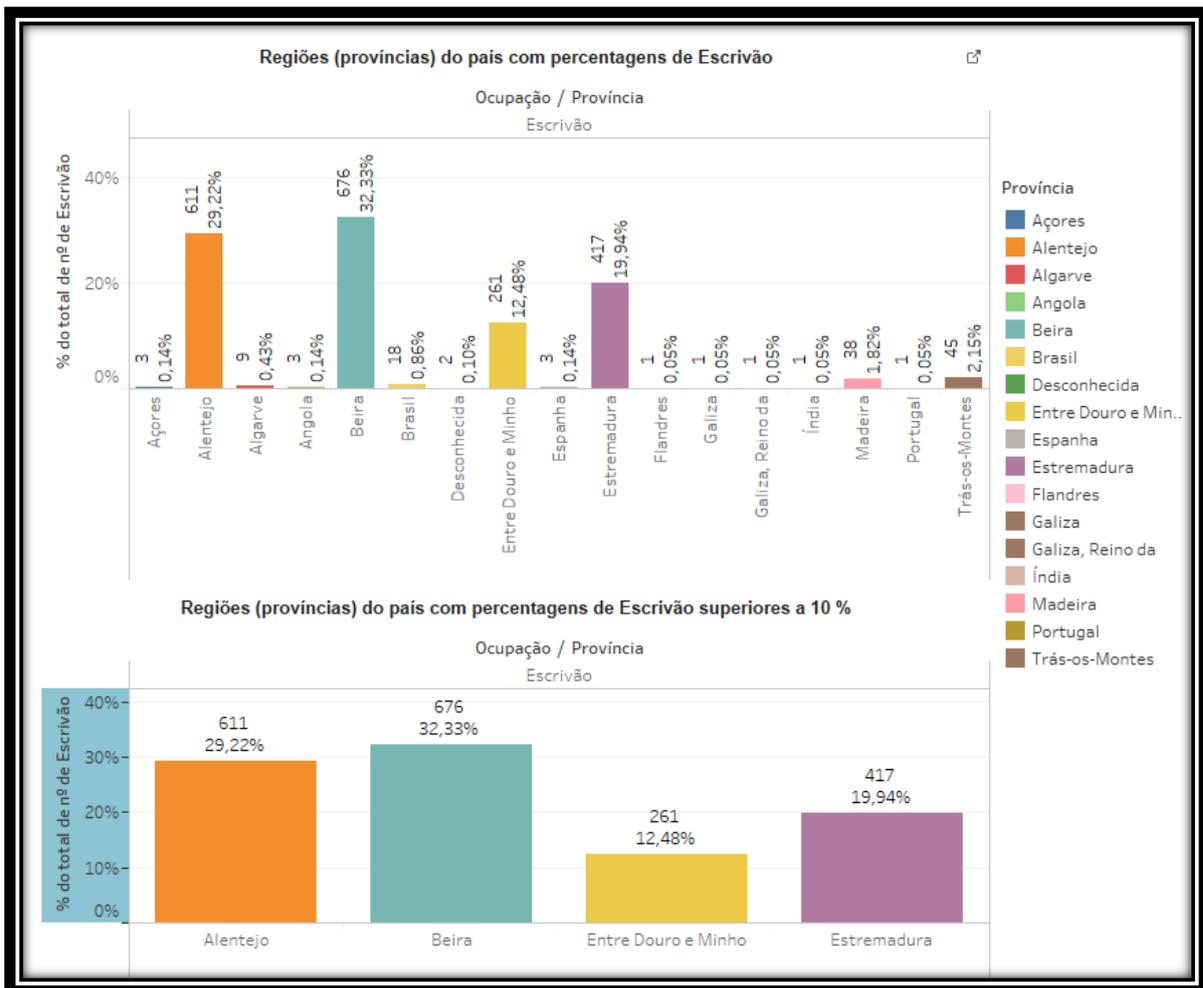
- Filtros em data**, são aplicados nos campos de data. O Tableau trata o campo de data de três maneiras diferentes durante a aplicação do campo de data. Pode aplicar o filtro tomando uma data relativa, em comparação aos dias de hoje, uma data absoluta ou intervalo de datas. Cada uma destas opções é apresentada quando um campo de data é arrastado para fora do painel de filtro. A Figura 41 apresenta as opções sobre os filtros em datas, tendo o utilizador um leque e variedade enorme de opções, podendo filtrar entre intervalo de duas datas, filtrar por dia, filtrar pelos dados de hoje, ou ontem ou amanhã, ou nos x dias atrás. Ainda é possível filtrar em semanas, meses, trimestres, semestre, anos, filtrar pelo início de uma data ou termino de uma data, ou programar uma função especial.

### 6.3. Resultados obtidos de análises de dados

Neste capítulo procurei exemplificar várias análises de dados, aplicando as técnicas de Business intelligence ao novo sistema prosopográfico de análise de relações e eventos, comparando a performance dos resultados obtidos, o tempo da execução, a qualidade de dados, entre outros requisitos.

Demonstrarei os resultados em diversas formas de visualização, registrando a importância do aplicativo tipo cubo, ao SPARES. Todos os resultados obtidos, são análises que no SPARES antigo não era possível realizar nem descobrir esses padrões de dados. Por isso, podemos afirmar sem dúvida que este projeto constitui uma mais valia para o CIDEHUS, nomeadamente para os investigadores e utilizadores do sistema prosopográfico de análise de relações e eventos.

## 6.3.1. Análises de dados - Caso A



**Figura 42 - Caso A – Província da idade moderna com as respetivas percentagens de escrivo**

De acordo com a Figura 42, verificamos que através de gráficos de barras obtemos a análise de percentagens de escrivo em todas as províncias registadas no sistema SPARES. Se observamos essa figura constatamos que dos casos estudados, a província da Beira tem maior número de escravos. Galiza, Flandres, Índia são as províncias que contabilizam menor número de registos.

Para obter esta análise, arrastamos na dimensão comissários o «Nome de P2» para marcas. Dado que essa coluna é qualitativa, é necessário mudar para contagem de medidas para ser quantitativa, e adicionar um cálculo que consiste em determinar a percentagem do total acumulado; depois arrastamos o campo província da dimensão «localidade».

No «Mostre Me» escolher tipo de visualização em barras, adicionar a coluna de província para cores para pintar o gráfico com diferentes cores conforme a percentagem de cada um. De seguida, haverá que adicionar ocupação no filtro; como a coluna é considerada

uma dimensão no filtro, já existe a lista de ocupações, pelo que só temos que seleccionar o escrivão.

Na segunda pesquisa, adicionou-se um filtro de medida que foi calculada, para apresentar somente os resultados superiores a 10 %, tendo dessa forma diminuído o número de províncias na pesquisa.

### 6.3.2. Análise de dados - Caso B

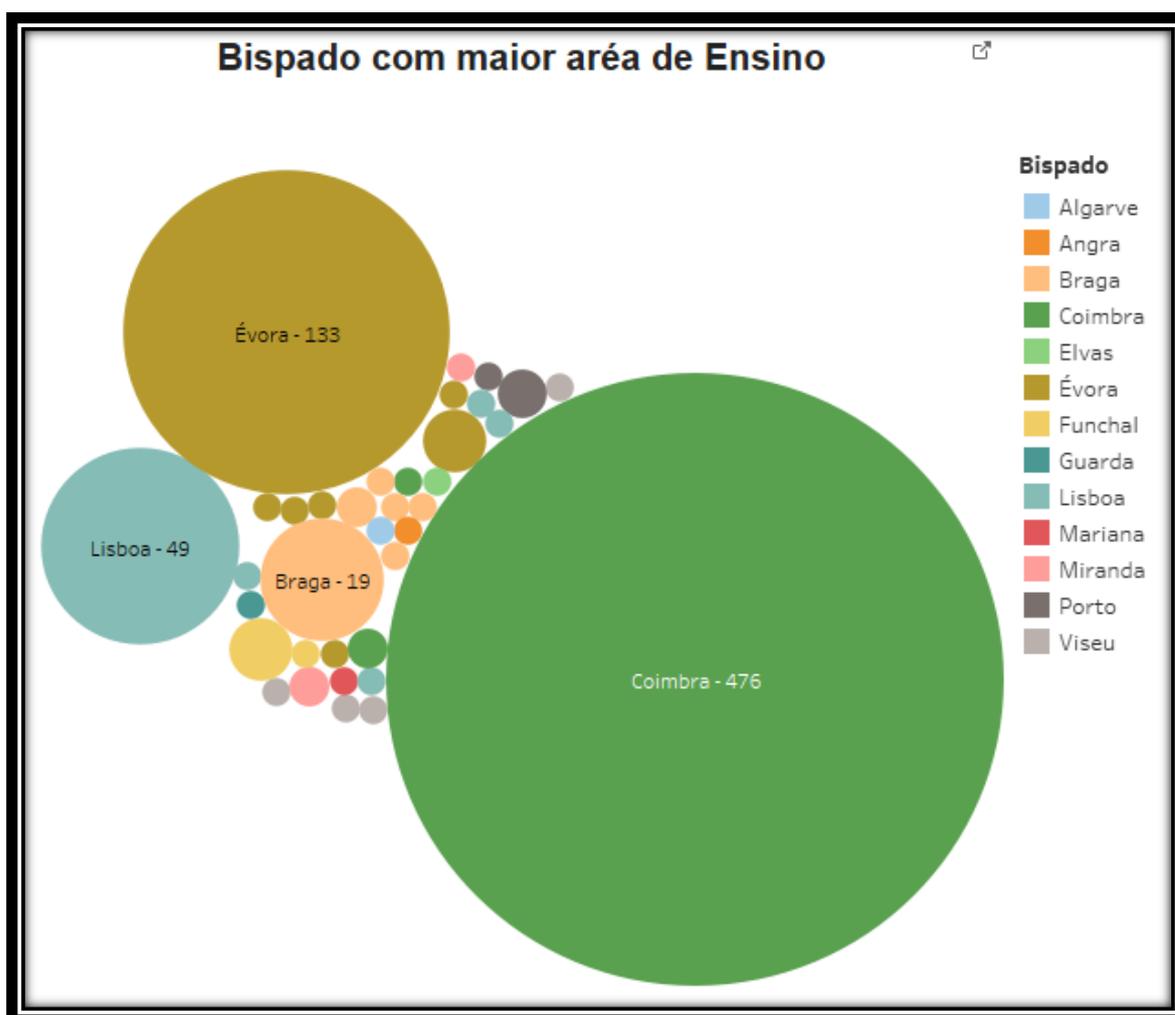
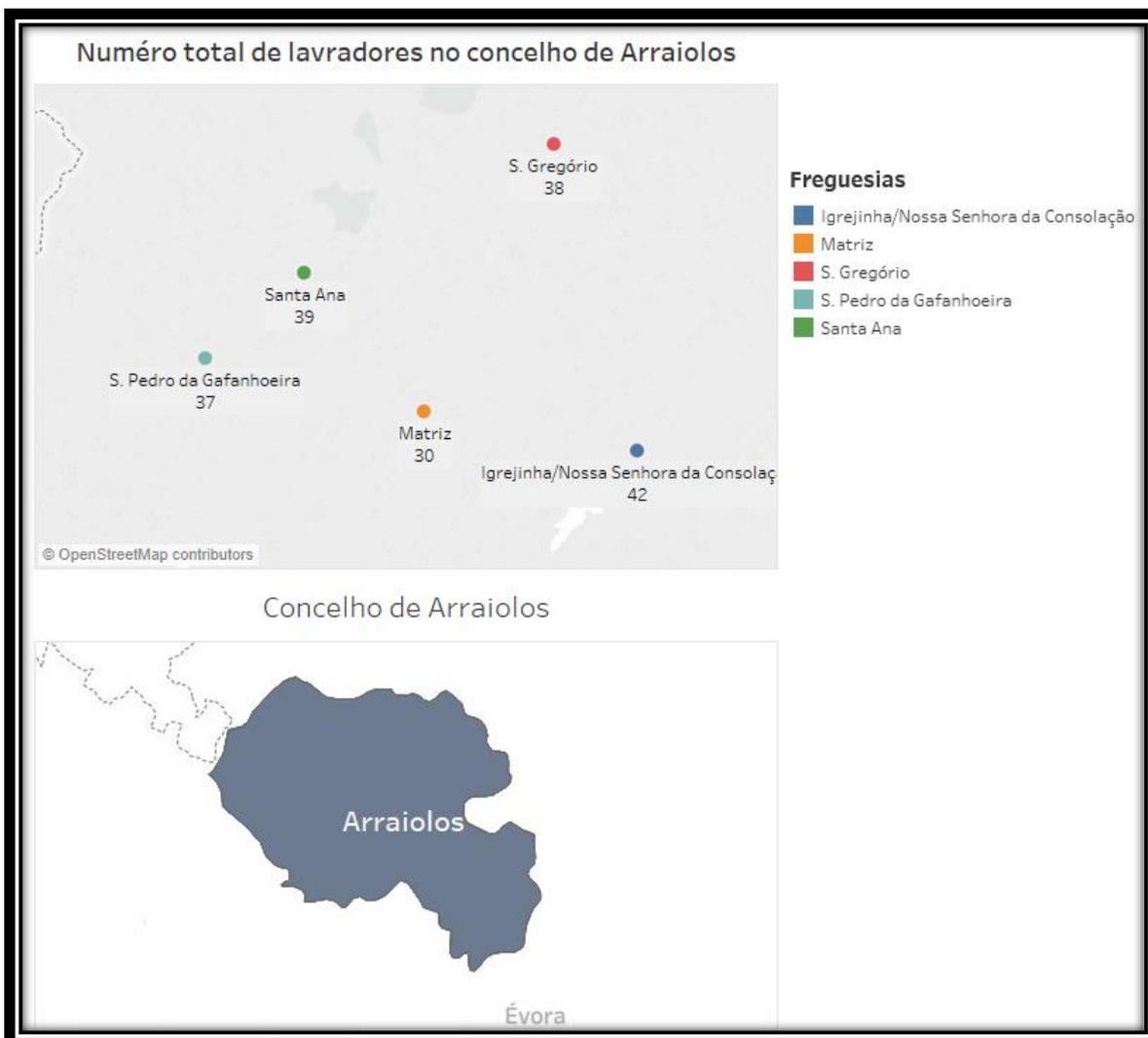


Figura 43 - Caso B – Bispados com maior área de Ensino

A Figura 43 apresenta os resultados em bolhas de pacotes, uma forma diferenciada de apresentar os dados. Essa análise consiste em analisar o bispado com maior área de ensino, não sendo surpreendente que a maioria absoluta seja o bispado de Coimbra.

Da mesma forma, na análise do caso B, verificamos que se trata de um caso bastante simples de realizar. Em primeiro lugar, arrastar o campo «Código de P2», e aplicar contagem, adicionar bispados, adicionar tipo setor no filtro e escolher a área de ensino, e por fim no «mostre me» escolher essa opção de apresentar os resultados em bolhas em pacotes.

### 6.3.3. Análises de dados - Caso C



**Figura 44 – Caso C - Análise obtidos número total de lavradores em cada freguesia de Arraiolos**

A Figura 44 apresenta os resultados em mapas de Idade Moderna, do concelho de Arraiolos. Essa análise, consiste em analisar em micro escalas o número de lavradores existente no concelho de Arraiolos. Nesse caso a análise parece complexa, mas é bastante simples. O primeiro passo consiste em adicionar a província do shape file à tela de exibição e criar o mapa com preenchimento. Primeiro haverá que arrastar o campo «Código de P2» e Concelho, e aplicar a contagem distinta de Código de P2.

Por fim, aplicam-se determinados filtros para ter essa pesquisa há que adicionar a coluna ocupação da dimensão ocupação nos filtros e escolher lavrador, arrastar o concelho no shape file e soltar nos filtros escolhendo o concelho de arraiolos, e no final na dimensão localidade arrastar o campo de freguesia e largar nas marcas, e adicionar a contagem em textos, para aparecer o número respetivo de lavradores em cima de qualquer freguesia. Nessa pesquisa, a freguesia de Igreja/Jinha tem o maior número de lavradores no concelho de Arraiolos, e a freguesia de Matrix é a que tem o menor número de lavradores.

### 6.3.4. Análise de dados - Caso D

Calcular o número e a percentagem total de Comissário de Santo Ofício em cada Província		
Província1	Nº total de comissários	% do total de Contagem de Comissários
Nulo	651	41,31%
Beira	255	16,18%
Entre-Douro-Minho	255	16,18%
Alentejo	172	10,91%
Estremadura	118	7,49%
Trás-os-Montes	63	4,00%
Brasil	30	1,90%
Algarve	13	0,82%
Açores	12	0,76%
Angola	2	0,13%
Madeira	2	0,13%
China	1	0,06%
Espanha	1	0,06%
Índia	1	0,06%
<b>Total geral</b>	<b>1 576</b>	<b>100,00%</b>

**Figura 45 – Caso D - Reporting de número e percentagem de comissários em cada província**

A Figura 45 contém os dados referente os número de comissários de Santo Ofício de cada província, e com a percentagem do total podemos concluir que a maioria dos casos não tem a localidade definida, sendo que cerca de 41% de dados sobre ocupação de comissários Santo Ofício não tem definido a província.

Esta mesma figura, evidencia que temos cerca de 1576 comissários, divididos por diversas províncias, ordenado de forma decrescente. A província com maior número de comissários de Santo Ofício é a Beira, a que se segue de Entre Minho e Douro e a seguir aparece o Alentejo com 172 comissários.

### 6.3.5. Análise de dados – Caso E

No caso E podemos observar um reporting de dados interessante, que na forma antiga de exploração do sistema SPARES, é impensável saber esses dados, mas com o Tableau é tudo muito intuitivo. Nessa análise de dados, é apresentado em cada século o número de pessoas que nasceram, nos finais de semanas e nos restantes dias. Constatamos que a maioria dos atores estudados nasceram no século XVIII, e o número menor ocorre no século XVII.

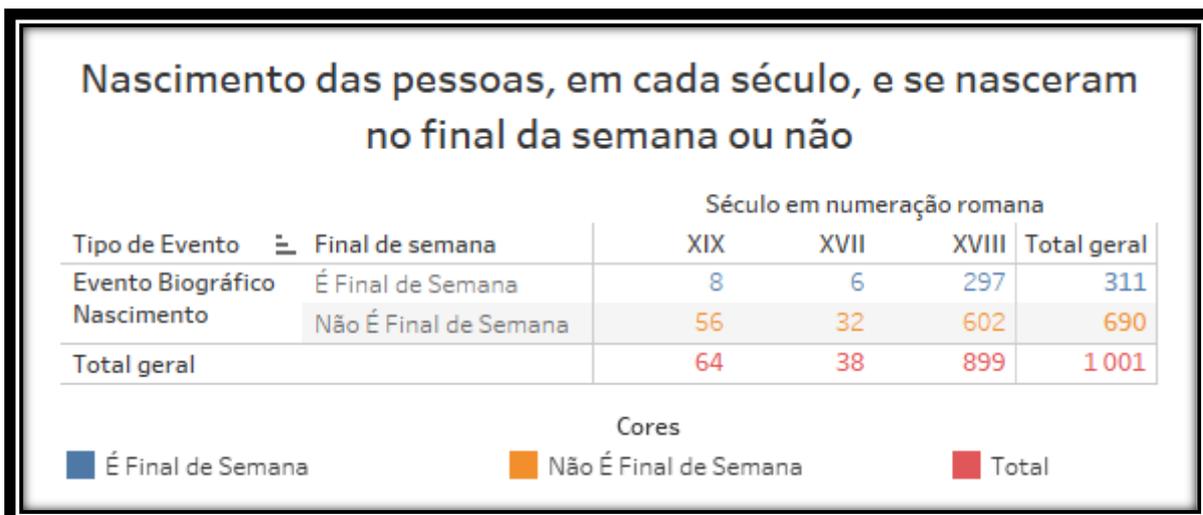


Figura 46 - Caso E - Reporting números de pessoas que nasceram em cada século

### 6.3.6. Análise de dados – Caso F

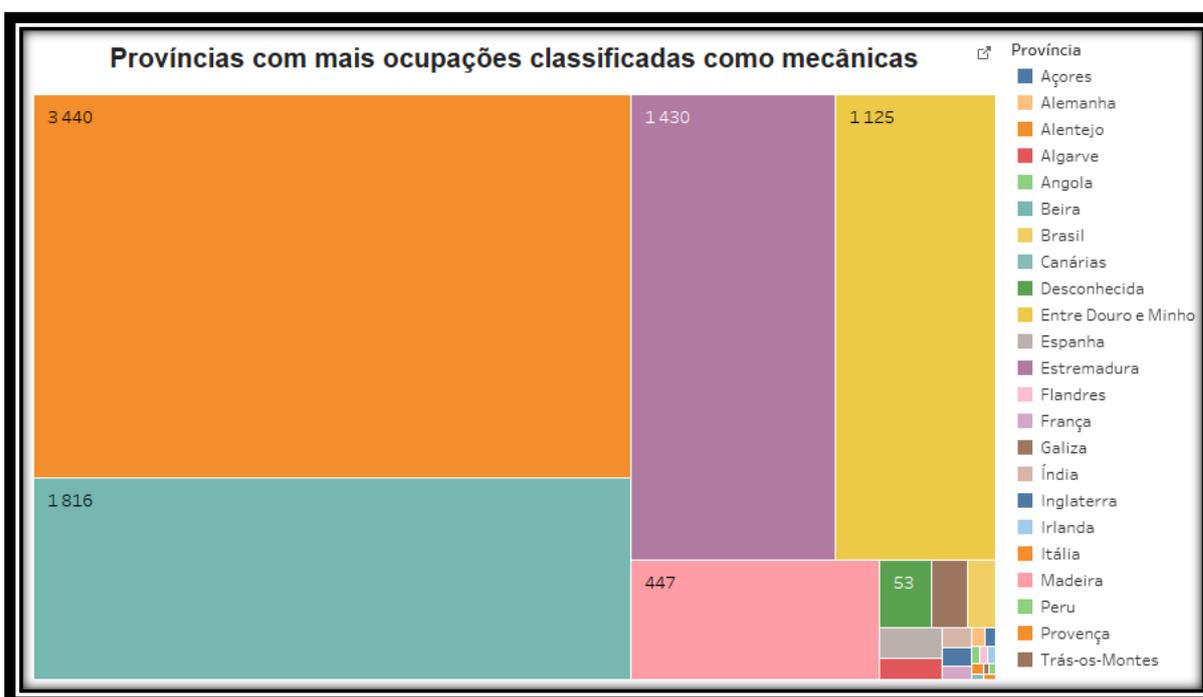
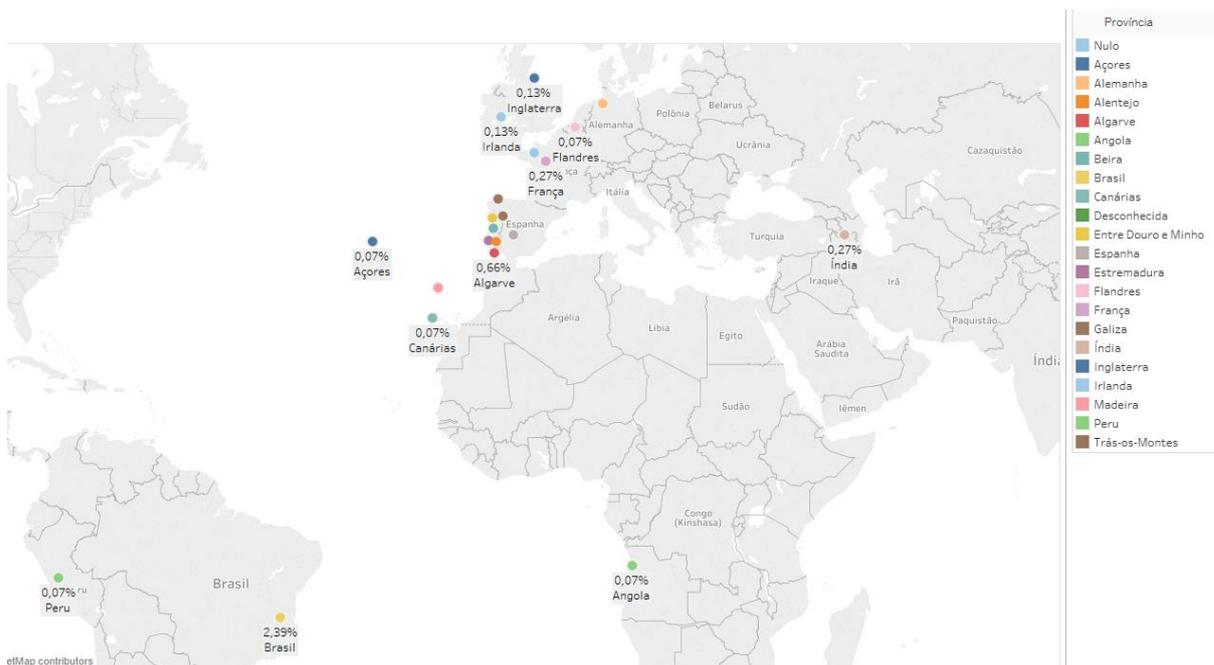


Figura 47 – Caso F – Províncias com os respetivos números total de ocupações classificadas como mecânicas

A Figura 47 mostra-nos um resultado com visualização diferente das outras; é efetuado através de mapas de árvores. Contabilizamos o número de ocupações classificadas como mecânicas, e conclui-se que a província de Alentejo tem cerca de 3440 ocupações registadas como mecânicas, seguida da Beira com 1816, e para completar o “pódio” aparece a Estremadura com cerca de 1430 registos.

### 6.3.7. Análises de dados - Caso G



**Figura 48 - Caso G – Províncias cartografado com a percentagem de tipo de ocupação Mercancia**

O caso G apresenta uma mapa mais generalizado de todas as províncias registadas com algum tipo de ocupação Mercancia. O Tableau tem o zoom do mapa para observamos os resultados de uma forma mais clara e concisa, dado que existem outras província no Brasil, Angola, Índia, Canárias, entre outras províncias, o que faz com que a imagem seja vista de uma forma muito pequena. Mas Portugal tem o maior número de províncias com o tipo de ocupação Mercancia.

# Capítulo 7

## **Conclusão e Estudos Futuros**



## 7. Conclusão e Estudos Futuros

Parafraseando Michael Porter de que *“Em qualquer empresa, a tecnologia da informação exerce efeitos poderosos sobre a vantagem competitiva, tanto no custo, quanto na diferenciação”* (Gestão, s.d), temos o cerne último do que representou a realização deste projeto em termos pessoais, porquanto o mesmo visa dotar uma organização da possibilidade de ter ao seu dispor, uma ferramenta informática, que de modo prático lhe permita ter acesso e um tratamento facilitado e fidedigno de dados.

O mote inicial deste trabalho, foi o enquadramento das necessidades de uma nova geração empresarial, totalmente comprometida com a competitividade, as exigências de tomada de decisões estratégicas, a que se alia a necessidade de uma análise preditiva Business Intelligence, que quando bem aplicada, pode impulsionar e maximizar o potencial de negócio em termos de lucro, retorno de investimento, coerência e coesão das organizações, independentemente da tipologia das mesmas.

Nesse emaranhado competitivo e de mudanças contínuas que as organizações enfrentam, a informação surge como um factor essencial para sua sobrevivência, mas não a informação em si, na sua forma puramente física e estática, mas antes a gestão da informação e do seu fluxo a fim de gerar conhecimento e poder assim oferecer um auxílio à tomada de decisão nas organizações.

Este trabalho permitiu por um lado, apresentar efetivamente um conjunto de técnicas e métodos de Business Intelligence que aplicados a uma base de dados prosopográfica – o sistema SPARES – permite efetivamente melhorar a forma atual de exploração de dados que é essencialmente baseada em filtros avulsos, o que se traduz num recurso importante para os investigadores. Por outro lado, o seu desenvolvimento, permitiu que pessoalmente tivesse a oportunidade de aprofundar os meus conhecimentos sobre ferramentas analíticas ligadas a data warehouse.

Com efeito, a conceção e elaboração deste projeto, permitiu não só a familiarização com os conjuntos de técnicas e métodos de Business Intelligence aplicados a uma rede prosopográfica, mas também com as novas ferramentas e técnicas utilizadas para a exploração e análise de dados no sistema SPARES, e com a Ecologia dos Dados que, segundo Caldeira (2011, p. 177), tem como fim *“a captação do funcionamento e da dinâmica das organizações, em que as pessoas são mediadores dos fluxos de informação. Na ecologia da informação, os dados são descritos e caracterizados de acordo com o modo como se relacionam com os subsistemas socioeconómicos da organização.”*

A pesquisa efetuada e os testes práticos de aplicabilidade, que podem ser identificados através dos estudos de caso, permitem afirmar que a implementação deste projeto garante, por si só, a possibilidade de se aumentar a qualidade de informação proporcionada aos investigadores e utilizadores do sistema SPARES. Por um lado, porque ao termos de analisar a informação existente na fonte de dados transacionais, isso permite-nos muitas vezes a identificação dos erros. Por outro lado, fornece informações confiáveis e consistentes que servem de base e orientação importante no processo estratégico da tomada de decisão.

Podemos agora com maior conhecimento, afirmar que o data warehouse é o meio e não o fim, pois o fim é o negócio da organização. A informação é um produto valioso e confere vantagem competitiva. A modelagem dimensional possui um grande potencial para as aplicações de data warehouse e os seus elementos, técnicas e passos que orientam o seu desenvolvimento, aumentam a probabilidade de sucesso do projeto.

A criação de um data warehouse utilizando a modelagem dimensional gera um suporte conveniente para a interface gráfica de sistema de business intelligence escolhida nesta dissertação: o Tableau. Esta aplicação permite que os utilizadores analisem os dados em diversas formas versus um amplo conjunto de perspetivas.

A dissertação de Mestrado ora apresentada, permitiu por fim, introduzir especificamente a ferramenta de data warehouse juntamente com as técnicas de Business Intelligence a um repositório de dados prosopográficos, em curso de desenvolvimento no centro de investigação CIDEHUS da Universidade de Évora.

Pode-se, em jeito de conclusão, afirmar categoricamente que o BI tem como objetivo principal transformar grandes quantidades de dados em informações de qualidade, para a tomada de decisão, de modo a possibilitar uma visão sistémica do negócio e auxiliar na distribuição uniforme dos dados entre os analistas de negócios. As aplicações de BI podem auxiliar em vários segmentos das organizações, sendo um auxílio valioso nas tendências e garantia da sustentabilidade e criação de valor das organizações.

É importante ressaltar que as soluções de BI acompanham as necessidades da organização, evoluindo paulatinamente, conforme vão surgindo novos recursos e desafios. O investimento em BI por parte das organizações é incentivado pela necessidade de criação de um ambiente proativo para a tomada de decisão.

## **7.1. Estudos Futuros**

Apesar do trabalho realizado ser promissor, surgem ainda alguns pontos que podem sempre ser aperfeiçoados. Como tarefas futuras, destacam-se uma série de sugestões, dado que em princípio o projeto apresentado poderá ter continuidade, de modo a poder oferecer novas faces ao cubo de dados para exploração, análise e descoberta de novos padrões de dados.

É notória e premente a preocupação futura em desenvolver um sistema de metadados, para integração de dados prosopográficos num silo destinado a Business Intelligence, optimizando-se a utilização de uma aplicação do tipo CUBO, para prospeção de dados no repositório desenvolvido.

Outra possível melhoria seria estudar a possibilidade efetiva de separar no sistema SPARES a parte dos sistemas transacionais dos sistemas analíticos. Criar um data warehouse destinada somente para análise de dados, formado por várias data marts, tendo por base sempre, as necessidades de análise dos utilizadores do sistema.

Ainda como melhoria, pode equacionar-se a criação de um data mart sobre genealogia, dado que é de vital importância relacionar dados de ocupação com as suas genealogias, principalmente saber a ocupação dos pais ou avós de um determinado ator.

São inesgotáveis as possibilidades de desenvolvimento e de promoção da melhoria contínua, a nível do objeto do presente projeto. Compete-nos a todos dar continuidade, para que os trabalhos propostos tenham aplicabilidade prática nas organizações e potencial de apoio futuro. Aqui fica o repto: dar às organizações ferramentas simples, práticas, confiáveis e que justifiquem o investimento das mesmas.



## 8. Referências

- Abacus, B. (s.d). *QlikMarket*. Consultado em 23 de Novembro de 2016. Disponível em: <http://market.qlik.com/stock-management-dashboard-in-qlikview.html>.
- Albino, N. (2009). *Estudo de Implementação de um Data Warehouse para uma Instituição do Ensino Superior*. Dissertação de Mestrado em Engenharia Informática. Escola de Ciências e Tecnologia - Universidade de Évora, Évora. 131 pp.
- Almeida, C. (2011). *A prosopografia ou biografia coletiva: limites, desafios e possibilidades*. Anais do XXVI Simpósio Nacional de História – ANPUH, São Paulo. Consultado em 14 de Fevereiro de 2017. Disponível em: [http://www.snh2011.anpuh.org/resources/anais/14/1300892678\\_ARQUIVO\\_anpuhsp2011.pdf](http://www.snh2011.anpuh.org/resources/anais/14/1300892678_ARQUIVO_anpuhsp2011.pdf).
- Antunes, C. (2008). *Data Mining e Data Warehousing da exploração de dados à descoberta de informação*. Departamento de Engenharia Informática, Instituto Superior Técnico, Lisboa. Consultado em 26 Fevereiro de 2017. Disponível em: <https://fenix.tecnico.ulisboa.pt/downloadFile/282093452010155/sad.v2.0.alunos.pdf>.
- Azevedo, A., e Caminha, D. (2013). *Prosopografia: possibilidades de aplicação do método no campo da administração*. IV Encontro de Ensino e Pesquisa em Administração e Contabilidade, Brasília. Consultado em 15 de Fevereiro de 2017. Disponível em: <http://www.anpad.org.br/admin/pdf/EnEPQ219.pdf>.
- Barbieri, C. (2001). *BI-business intelligence: modelagem e tecnologia*. Axcel Books.
- Barbosa, F., e Oliveira, K. (2007). *Transformação de Dados para Data Warehouse*. Monografia de Bacharelato em Sistemas de Informação. Universidade Católica, Brasília. 54 pp. Consultado em 28 de Fevereiro de 2017. Disponível em: [http://oasisbr.ibict.br/vufind/Record/UCB\\_ca3c7dcad68a8276cd34334b1c6390cc](http://oasisbr.ibict.br/vufind/Record/UCB_ca3c7dcad68a8276cd34334b1c6390cc).
- Bonel, C. (2015). *Afinal, O Que é Business Intelligence?* Rio de Janeiro: 1ª Edição. Clube de Autores.
- Braghittoni, R. (2017). *Business Intelligence: Implementar do jeito certo e a custo zero*. Casa do Código.
- Bruzarosco, D., Castoldi, A., e Pacheco, R. (2000). *Criando data warehouse com o modelo dimensional*. Acta Scientiarum 22(5):1389-1397. Consultado em 10 de Junho de 2017. Disponível em: [https://www.researchgate.net/profile/Roberto\\_Pacheco/publication/277203169\\_Criando\\_o\\_data\\_warehouse\\_com\\_o\\_modelo\\_dimensional/links/568d1e7708aeecf87b218304/Criando\\_o\\_data\\_warehouse\\_com\\_o\\_modelo\\_dimensional/links/568d1e7708aeecf87b218304/Criando\\_o\\_data\\_warehouse\\_com\\_o\\_modelo\\_dimensional.pdf](https://www.researchgate.net/profile/Roberto_Pacheco/publication/277203169_Criando_o_data_warehouse_com_o_modelo_dimensional/links/568d1e7708aeecf87b218304/Criando_o_data_warehouse_com_o_modelo_dimensional.pdf).
- Bulst, N. (2005). *Sobre o objeto e o método da prosopografia*. Politeia: história e sociedade, volume 05, n. 1, p. 47-67. Consultado em 15 de Fevereiro de 2017. Disponível em: <http://periodicos.uesb.br/index.php/politeia/article/viewFile/190/211>.
- Caldeira, C. (2011). *A Arte das Bases de Dados*. Lisboa: Edições Sílabo, Lda.
- Caldeira, C. (2012). *Data Warehousing - Conceitos e Modelos*. Lisboa: Edições Sílabo, Lda.

- Ciferri, C., e Ciferri, R. (s.d). *Modelagem Multidimensional*. Consultado em 09 de Maio de 2017. Disponível em: <http://wiki.icmc.usp.br/images/6/6a/SCC5911-02-ModelagemMultidimensional.pdf>.
- Codd, E., Codd, S., e Salley, C. (1993). *Providing OLAP to User-Analysts: An IT Mandate*. Consultado em 24 de Abril de 2017. Disponível em: [http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem\\_dwh/lit/Cod93.pdf](http://www.minet.uni-jena.de/dbis/lehre/ss2005/sem_dwh/lit/Cod93.pdf).
- Costa, S. (2012). *Sistema de Business Intelligence como suporte à Gestão Estratégica*. Tese de Mestrado em Engenharia e Gestão de Sistemas de Informação. Escola de Engenharia - Universidade de Minho, Braga. 135 pp. Consultado em 4 de Abril de 2017. Disponível em: <http://repositorium.sdum.uminho.pt/handle/1822/25810>.
- Costa, S. (2014). *Sistemas de Informação Dashboards*. Consultado em 05 de Abril de 2017. Disponível em: [https://edisciplinas.usp.br/pluginfile.php/188607/mod\\_resource/content/1/Dashboards.pdf](https://edisciplinas.usp.br/pluginfile.php/188607/mod_resource/content/1/Dashboards.pdf).
- Dalfovo, O., e Tamborlim, N. (2017). *Business Intilligence. Estudos e casos: na Gestão de Tecnologia da Informação como Inteligência nos Negócios*. 1ª edição. Blumenau: Edição Clube de Autores.
- Elias, D. (s.d.). *O que significa OLTP e OLAP na prática?* Canaltech. Consultado em 04 de Abril de 2017. Disponível em: <https://corporate.canaltech.com.br/o-que-e/business-intelligence/o-que-significa-oltp-e-olap-na-pratica/>.
- Ferreira, A. (2016). *OntoSPARES: da linguagem natural às ontologias. Contributos para a classificação automática de dados históricos (séc. XVI-XVIII)*. Tese Doutoramento em Informática. Escola de Ciências e Tecnologia, Departamento de Informática - Universidade de Évora, Évora. 111 pp.
- Ferreira, A., Caldeira, C., e Olival, F. (2012). *Dados Qualitativos, Prosopografia e Análise de Redes*. Instituto de Ciências Sociais - Universidade de Lisboa. Consultado em 22 de Novembro de 2016. Disponível em: <http://dspace.uevora.pt/rdpc/handle/10174/9682>.
- Ferreira, J., Miranda, M., Abelha, A., e Machado, J. (2010). *O Processo ETL em Sistemas Data Warehouse*. INForum 2010 - II Simpósio de Informática, Luís S. Barbosa, Miguel P. Correia, (eds), 9-10 Setembro, 2010, pp. 757–765. Universidade do Minho, Braga. Consultado em 10 de Abril de 2017. Disponível em: <http://repositorium.sdum.uminho.pt/handle/1822/11435>.
- Ferreira, N. (2012). *Análise exploratória de hierarquias em base de dados multidimensionais*. Tese de Mestrado em Análise de Dados e Sistemas de Apoio à Decisão. Faculdade de Economia - Universidade de Porto, Porto. 85 pp. Consultado em 4 de Junho de 2017. Disponível em: <https://repositorio-aberto.up.pt/bitstream/10216/73510/2/25383.pdf>.
- Filho, H., Clericuzi, A., Souza, K., e Bione, B. (2011). *Dados qualitativos, prosopografia e análise de redes*. VIII SEGeT – Simpósio de Excelência em Gestão e Tecnologia . Consultado em 19 de Fevereiro de 2017. Disponível em: <http://www.aedb.br/seget/arquivos/artigos11/32514469.pdf>.

- Garrido, Á., Costa, L., e Duarte, L. (2012). *Estudos em Homagem a Joaquim romero Magalhães Economia, Instituições e Império*. Coimbra: Edições Almedina, S.A.
- Gestão, P. (s.d). *As Estratégias Genéricas de Michel Porter*. Consultado em 13 de Junho de 2017. Disponível em: <https://www.portal-gestao.com/artigos/6710-as-estrat%C3%A9gias-gen%C3%A9ricas-de-michael-porter.html>.
- Gibson, S. (2017). *Gartner Magic Quadrant For Business Intelligence 2017 – Cloud is Coming(Slowly)*. Consultado 9 de Maio de 2017. Disponível em: <http://optimalbi.com/blog/2017/02/17/gartner-magic-quadrant-for-business-intelligence-2017-cloud-is-coming-slowly/>.
- Gomes, M. (2011). *Modelação de um data warehouse para a Direcção Geral do Tesouro e Finanças e implementação de um data mart para o processo de gestão patrimonial*. Mestrado em Estatística e Gestão de Informação. Instituto Superior de Estatística e Gestão da Informação - Universidade Nova de Lisboa, Lisboa. 88 pp. Consultado em 13 de Fevereiro de 2017. Disponível: em <https://run.unl.pt/handle/10362/5366>.
- Gouveia, A. R. (2013). *Solução de Business Para Seguros*. Dissertação de Mestrado em Engenharia de Redes e Sistemas Informático. Faculdade de Ciências - Universidade do Porto, Porto. 96 pp. Consultado em 05 de Maio de 2017. Disponível em: [https://sigarra.up.pt/fcup/pt/pub\\_geral.pub\\_view?pi\\_pub\\_base\\_id=24521](https://sigarra.up.pt/fcup/pt/pub_geral.pub_view?pi_pub_base_id=24521).
- Huijbers, C. (2011). *Gartner Magic Quadrant for Business Intelligence Platforms 2011*. Consultado em 6 de Junho de 2017. Disponível em: <https://clinthuijbers.wordpress.com/2011/02/14/gartner-magic-quadrant-for-business-intelligence-platforms-2011-microsoft-is-the-leader/>.
- Kimball, R., e Ross, M. (2002). *The Data Warehouse Toolkit. the complete guide to dimensional modeling*. New York: Jonh Wiley & Sons, Inc.
- Kimball, R., e Ross, M. (2013). *The Data Warehouse Toolkit:The Definitive Guide to Dimensional Modeling*. 3rd Edition. Indianapolis: Wiley.
- Lima, L. S. (s.d). *Porque tomar decisões utilizando Business Intelligence*. DEVMEDIA. Consultado em 26 de Março de 2017. Disponível em: <http://www.devmedia.com.br/porque-tomar-decisoes-utilizando-business-intelligence/28341>.
- Microsoft. (2010). *A concepção da dimensão data num sistema de BI - Parte I*. Consultado em 21 de Fevereiro de 2017. Disponível em: <https://support.microsoft.com/pt-pt/help/2293691>.
- Murillo, M. (2016). *Agile Data Warehousing and Business Intelligence in Action*. Consultado em 14 de Março de 2016. Disponível em: de <https://www.thoughtworks.com/insights/blog/agile-data-warehousing-and-business-intelligence-action>.
- Nicolet, C. (1970). *Prosopographie et histoire sociale: Rome et Italie à l'époque républicaine. Annales: économie, sociétés, civilisations*. Paris, v. 25, n. 5, p. 1209-1228. Consultado em 14 de fevereiro de 2017. Disponível em: <http://www.persee.fr/doc/ah>.
- Olival, F. (2012). *Comissários das Ordens Militares e comissários do Santo Ofício em Portugal: dois modelos de actuação*. As Ordens Militares: freires, guerreiros,

- cavaleiros: actas do VI Encontro sobre Ordens Militares, coord. Isabel Cristina Fernandes, Vol. I, Palmela: Município de Palmela / GESOS. Consultado em 20 de Março de 2017. Disponível em: <http://dspace.uevora.pt/rdpc/handle/10174/7479>.
- Oracle, C. (s.d). *Data Warehousing and OLAP*. Consultado em 23 de Novembro de 2016. Disponível em: <http://www.oracle.com/technetwork/articles/sql/11g-dw-olap-100058.html>.
- Ornai, A. (2014). *Das Bases de Dados aos Sistemas de Apoio à Decisão: Ensaio de Aplicação a uma Instituição do Ensino*. Dissertação de Mestrado em Engenharia Informática. Escola de Ciências e Tecnologia - Universidade de Évora, Évora. 53 pp. Consultado em 5 de Junho de 2017. Disponível em: <http://dspace.uevora.pt/rdpc/bitstream/10174/12997/1/Disserta%C3%A7%C3%A3o%20de%20Mestrado.pdf>.
- Primark, F. (2008). *Decisões com BI (Business Intelligence)*. Ciência Moderna.
- Quintanilha, S., e Moraes, T. (s.d). *Vantagens e desvantagens do “Business Intelligence”, como forma de projetar a inteligência nos negócios de pequenas e médias empresas*. Consultado em 20 de Abril de 2017. Disponível em: [http://www.recap.eng.uerj.br/lib/exe/fetch.php?id=2009&cache=cache&media=recap\\_2009-10.pdf](http://www.recap.eng.uerj.br/lib/exe/fetch.php?id=2009&cache=cache&media=recap_2009-10.pdf).
- Rainardi, V. (2008). *Building a Data Warehouse - With Examples in SQL Server*. New York: Apress.
- Santos, M., e Ramos, I. (2006). *Business Intelligence : tecnologias da informação na gestão de conhecimento*. FCA - Editora de Informática.
- Sezões, C., Oliveira, J., e Baptista, M. (2006). *Business Intelligence*. Porto: SPI – Sociedade Portuguesa de Inovação.
- Souza, N. A. (s.d). *O que é Business Intelligence (BI)?* . Consultado em 14 de Março de 2017. Disponível em: <http://www.raffcom.com.br/blog/o-que-e-business-intelligence-bi/>.
- Stone, L. (1971). *Prosopography*. Daedalus: journal of American Academy of Arts and Sciences, vol. 100, nº 1, p. 46-79. Consultado em 15 de Fevereiro de 2017. Disponível em: [http://isites.harvard.edu/fs/docs/icb.topic1525486.files/Stone\\_Prosopography.pdf](http://isites.harvard.edu/fs/docs/icb.topic1525486.files/Stone_Prosopography.pdf).
- Tableau. (s.d). *Conceitos do Tableau*. Consultado em 24 de Maio de 2017. Disponível em: <http://onlinehelp.tableau.com/current/pro/desktop/pt-br/concepts.html>.
- Tableau. (s.d). *Missão do Tableau*. Consultado em 29 de Maio de 2017. Disponível em: <https://www.tableau.com/pt-br/about/mission>.
- w3ii. (s.d). *Tableau Tutorial*. Consultado em 23 de Maio de 2017. Disponível em: <http://www.w3ii.com/pt/tableau/default.html>.

# Anexos

## A. Script de criação da dimensão Data (SQL-Server)

```
DROP PROCEDURE IF EXISTS [inserir_dados_data];
DROP TABLE IF EXISTS [data];

CREATE TABLE [data] (
    "Código da data" INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    "Data" VARCHAR (50),
    "Descrição completa da data" TEXT,
    "Descrição abreviada da data" TEXT,
    "Ano em número" SMALLINT,
    "Mês em número" SMALLINT,
    "Dia em número" SMALLINT,
    "Dia de Semana em número" SMALLINT,
    "Semestre em número" SMALLINT,
    "Nome de semestre" VARCHAR(20),
    "Trimestre em número" SMALLINT,
    "Nome de trimestre" VARCHAR(20),
    "Ano bissexto" VARCHAR(20) ,
    "Nome do mês" VARCHAR(15),
    "Abreviatura do mês" CHAR(3),
    "Nome de dia" VARCHAR(20),
    "Abreviatura do dia" CHAR(3),
    "Século em Número" INT,
    "Século em Numeração Romana" VARCHAR(8),
    "Década em Número" CHAR(2),
    "Década por Extenso" VARCHAR(50),
    "Número da semana no ano" INT,
    "Numero da semana do mês" INT,
    "Estação do ano" VARCHAR(10),
    "Final de semana" TEXT,
    "Feriado" CHAR(3),
    "Nome do Feriado" VARCHAR(50),
    "Dia útil" TEXT,
    "SQL-data" DATE
);

-----

GO
CREATE PROCEDURE [inserir_dados_data](@dataInicial DATE, @dataFinal DATE)
AS
BEGIN
    DECLARE @date VARCHAR(50),
            @data DATE,
            @descricaoCompletaDaData VARCHAR(100),
            @descricaoAbreviadaDaData VARCHAR(100),
            @anoEmNúmero SMALLINT,
            @mêsEmNúmero SMALLINT,
            @diaEmNúmero SMALLINT,
            @diaDeSemanaEmNúmero SMALLINT,
            @SemestreEmNúmero SMALLINT,
            @nomeDeSemestre VARCHAR(20),
            @trimestreEmNúmero SMALLINT,
            @nomeDeTrimestre VARCHAR(20),
            @anoBissexto VARCHAR(20),
            @nomeDoMês VARCHAR(15),
            @abreviaturaDoMês CHAR(3),
            @nomeDeDia VARCHAR(20),
            @abreviaturaDoDia CHAR(3),
            @séculoEmNúmero INT,
            @séculoEmNumeraçãoRomana VARCHAR(8),
            @decadaEmNúmero CHAR(2),
            @decadaPorExtenso VARCHAR(50),
            @numeroDaSemanaNoAno INT,
```

```

@numeroDaSemanaDoMês INT,
@estaçãoDoAno VARCHAR(10),
@finalDeSemana VARCHAR(20),
@feriado CHAR(3),
@nomeDoFeriado VARCHAR(50),
@diaUtil VARCHAR(20),
@SqlData DATE,
@temp VARCHAR;

WHILE @dataInicial <= @dataFinal
BEGIN
SET @data = @dataInicial
SET @date = CAST(@dataInicial AS VARCHAR)
SET @anoEmNúmero = YEAR(@data)
SET @mêsEmNúmero = MONTH(@data)
SET @diaEmNúmero = DAY(@data)
SET @diaDeSemanaEmNúmero = DATEPART(WEEKDAY,@data)

SET @trimestreEmNúmero = CASE WHEN @mêsEmNúmero in (1,2,3) THEN 1
                                WHEN @mêsEmNúmero in (4,5,6) THEN 2
                                WHEN @mêsEmNúmero in (7,8,9) THEN 3
                                ELSE 4 END;

SET @nomeDeTrimestre = CASE
    WHEN @trimestreEmNúmero = 1 THEN 'Primeiro Trimestre'
    WHEN @trimestreEmNúmero = 2 THEN 'Segundo Trimestre'
    WHEN @trimestreEmNúmero = 3 THEN 'Terceiro Trimestre'
    ELSE 'Quarto Trimestre' END;

SET @SemestreEmNúmero = CASE WHEN @trimestreEmNúmero in (1,2) THEN 1
                                ELSE 2 END;

SET @nomeDeSemestre = CASE
    WHEN @SemestreEmNúmero = 1 THEN 'Primeiro Semestre'
    ELSE 'Segundo Semestre' END;

SET @anoBissexto = CASE
    WHEN (@anoEmNúmero % 4) = 0 THEN 'É Ano Bissexto'
    ELSE 'Não é ano Bissexto' END;

SET @nomeDoMês = CASE
    WHEN @mêsEmNúmero = 1 THEN 'Janeiro'
    WHEN @mêsEmNúmero = 2 then 'Fevereiro'
    WHEN @mêsEmNúmero = 3 then 'Março'
    WHEN @mêsEmNúmero = 4 then 'Abril'
    WHEN @mêsEmNúmero = 5 then 'Maio'
    WHEN @mêsEmNúmero = 6 then 'Junho'
    WHEN @mêsEmNúmero = 7 then 'Julho'
    WHEN @mêsEmNúmero = 8 then 'Agosto'
    WHEN @mêsEmNúmero = 9 then 'Setembro'
    WHEN @mêsEmNúmero = 10 then 'Outubro'
    WHEN @mêsEmNúmero = 11 then 'Novembro'
    WHEN @mêsEmNúmero = 12 then 'Dezembro' END

SET @abreviaturaDoMês = CASE
    WHEN @mêsEmNúmero = 1 then 'Jan'
    WHEN @mêsEmNúmero = 2 then 'Fev'
    WHEN @mêsEmNúmero = 3 then 'Mar'
    WHEN @mêsEmNúmero = 4 then 'Abr'
    WHEN @mêsEmNúmero = 5 then 'Mai'
    WHEN @mêsEmNúmero = 6 then 'Jun'
    WHEN @mêsEmNúmero = 7 then 'Jul'
    WHEN @mêsEmNúmero = 8 then 'Ago'
    WHEN @mêsEmNúmero = 9 then 'Set'
    WHEN @mêsEmNúmero = 10 then 'Out'
    WHEN @mêsEmNúmero = 11 then 'Nov'
    WHEN @mêsEmNúmero = 12 then 'Dez' END

SET @nomeDeDia = CASE
    WHEN @diaDeSemanaEmNúmero = 1 THEN 'Domingo'

```

```

        WHEN @diaDeSemanaEmNúmero = 2 then 'Segunda-Feira'
        WHEN @diaDeSemanaEmNúmero = 3 then 'Terça-Feira'
        WHEN @diaDeSemanaEmNúmero = 4 then 'Quarta-Feira'
        WHEN @diaDeSemanaEmNúmero = 5 then 'Quinta-Feira'
        WHEN @diaDeSemanaEmNúmero = 6 then 'Sexta-Feira'
        ELSE 'Sábado' END

SET @abreviaturaDoDia = CASE
        WHEN @diaDeSemanaEmNúmero = 1 THEN 'Dom'
        WHEN @diaDeSemanaEmNúmero = 2 then 'Seg'
        WHEN @diaDeSemanaEmNúmero = 3 then 'Ter'
        WHEN @diaDeSemanaEmNúmero = 4 then 'Qua'
        WHEN @diaDeSemanaEmNúmero = 5 then 'Qui'
        WHEN @diaDeSemanaEmNúmero = 6 then 'Sex'
        ELSE 'Sab' END

SET @numeroDaSemanaNoAno = DATEPART(wk,@data);

SET @numeroDaSemanaDoMês = CASE
        WHEN @diaEmNúmero < 8 THEN 1
        WHEN @diaEmNúmero < 15 THEN 2
        WHEN @diaEmNúmero < 22 THEN 3
        WHEN @diaEmNúmero < 29 THEN 4
        ELSE 5 END

IF @diaDeSemanaEmNúmero in(1,7)
    SET @finalDeSemana = 'É Final de Semana';
ELSE
    SET @finalDeSemana = 'Não É Final de Semana';

IF @data between CAST(CONVERT(CHAR(4),@anoEmNúmero)+'/09/23' AS DATE)
and CAST(CONVERT(CHAR(4),@anoEmNúmero)+'/12/20' AS DATE)
    SET @estaçãoDoAno = 'Outono'
ELSE IF @data between CAST(CONVERT(CHAR(4),@anoEmNúmero)+'/03/21' AS
DATE) and CAST(CONVERT(CHAR(4),@anoEmNúmero)+'/06/20' AS DATE)
    SET @estaçãoDoAno = 'Primavera'
ELSE IF @data between CAST(CONVERT(CHAR(4),@anoEmNúmero)+'/06/21' AS
DATE) and CAST(CONVERT(CHAR(4),@anoEmNúmero)+'/09/22' AS DATE)
    SET @estaçãoDoAno = 'Verão'
ELSE
    SET @estaçãoDoAno = 'Inverno'

SET @séculoEmNúmero = (@anoEmNúmero / 100)+1;

SET @séculoEmNumeraçãoRomana = CASE
        WHEN @séculoEmNúmero = 1 THEN 'I'
        WHEN @séculoEmNúmero = 2 then 'II'
        WHEN @séculoEmNúmero = 3 then 'III'
        WHEN @séculoEmNúmero = 4 then 'IV'
        WHEN @séculoEmNúmero = 5 then 'V'
        WHEN @séculoEmNúmero = 6 then 'VI'
        WHEN @séculoEmNúmero = 7 then 'VII'
        WHEN @séculoEmNúmero = 8 then 'VIII'
        WHEN @séculoEmNúmero = 9 then 'IX'
        WHEN @séculoEmNúmero = 10 then 'X'
        WHEN @séculoEmNúmero = 11 then 'XI'
        WHEN @séculoEmNúmero = 12 then 'XII'
        WHEN @séculoEmNúmero = 13 then 'XIII'
        WHEN @séculoEmNúmero = 14 then 'XIV'
        WHEN @séculoEmNúmero = 15 then 'XV'
        WHEN @séculoEmNúmero = 16 then 'XVI'
        WHEN @séculoEmNúmero = 17 then 'XVII'
        WHEN @séculoEmNúmero = 18 then 'XVIII'
        WHEN @séculoEmNúmero = 18 then 'XIX'
        WHEN @séculoEmNúmero = 20 then 'XX'
        ELSE 'XXI' END

SET @temp = SUBSTRING(CAST(@anoEmNúmero AS VARCHAR(4)),3,1)
SET @decadaEmNúmero= @temp+'0'

```

```

SET @decadaPorExtenso = 'Década de ' + @decadaEmNúmero + ' do Século '
+cast(@séculoEmNumeraçãoRomana as varchar);

IF (@mêsEmNúmero = 1 and @diaEmNúmero = 1)
    OR (@mêsEmNúmero = 2 and @diaEmNúmero = 28)
    OR (@mêsEmNúmero = 5 and @diaEmNúmero = 1)
    OR (@mêsEmNúmero = 6 and @diaEmNúmero = 10)
    OR (@mêsEmNúmero = 11 and @diaEmNúmero = 1)
    OR (@mêsEmNúmero = 12 and @diaEmNúmero = 25))
    SET @feriado = 'Sim';
ELSE
    SET @feriado = 'Não';

IF @finalDeSemana = 'É Final de Semana' OR @feriado = 'Sim'
    SET @diaUtil = 'Não é dia Útil'
ELSE
    SET @diaUtil = 'É dia Útil';

SET @SqlData = CONVERT(DATE, @data);

IF (@mêsEmNúmero = 1 and @diaEmNúmero = 1)
    SET @nomeDoFeriado = 'Ano Novo'
ELSE IF (@mêsEmNúmero = 2 and @diaEmNúmero = 28)
    SET @nomeDoFeriado = 'Carnaval'
ELSE IF (@mêsEmNúmero = 5 and @diaEmNúmero = 1)
    SET @nomeDoFeriado = 'Dia do Trabalhador'
ELSE IF (@mêsEmNúmero = 6 and @diaEmNúmero = 10)
    SET @nomeDoFeriado = 'Dia de Portugal'
ELSE IF (@mêsEmNúmero = 11 and @diaEmNúmero = 1)
    SET @nomeDoFeriado = 'Dia de Todos os Santos'
ELSE IF (@mêsEmNúmero = 12 and @diaEmNúmero = 25)
    SET @nomeDoFeriado = 'Natal'
ELSE
    SET @nomeDoFeriado = null;

SET @descriçãoCompletaDaData = @nomeDeDia + ', ' + CAST(@diaEmNúmero AS
VARCHAR) + ' de ' + CAST(@nomeDoMês AS VARCHAR) + ' de ' + CAST(@anoEmNúmero AS
VARCHAR);
SET @descriçãoAbreviadaDaData = @abreviaturaDoDia + ', ' +
CAST(@diaEmNúmero AS VARCHAR) + ' de ' + CAST(@abreviaturaDoMês AS VARCHAR) + ' de
' + CAST(@anoEmNúmero AS VARCHAR);

```

-----Inserir a data Completa-----

```

INSERT INTO [data]
SELECT
    @date,
    @descriçãoCompletaDaData,
    @descriçãoAbreviadaDaData,
    @anoEmNúmero,
    @mêsEmNúmero,
    @diaEmNúmero,
    @diaDeSemanaEmNúmero,
    @SemestreEmNúmero,
    @nomeDeSemestre,
    @trimestreEmNúmero,
    @nomeDeTrimestre,
    @anoBissexto,
    @nomeDoMês,
    @abreviaturaDoMês,
    @nomeDeDia,
    @abreviaturaDoDia,
    @séculoEmNúmero,
    @séculoEmNumeraçãoRomana,
    @decadaEmNúmero,
    @decadaPorExtenso,
    @numeroDaSemanaNoAno,
    @numeroDaSemanaDoMês,
    @estaçãoDoAno,
    @finalDeSemana,
    @feriado,

```

```

        @nomeDoFeriado,
        @diaUtil,
        @sqlData;
-----Inserir os dados da Data Completo,mas antes dssa DATA -----

        INSERT INTO [data]
        SELECT
            @date,
            'Sabe-se que é antes de, ' + CAST(@diaEmNúmero AS
VARIABLE) + ' de ' + CAST(@nomeDoMês AS VARIABLE) + ' de ' + CAST(@anoEmNúmero AS
VARIABLE),
            'Sabe-se que é antes de, ' + CAST(@diaEmNúmero AS
VARIABLE) + ' de ' + CAST(@abreviaturaDoMês AS VARIABLE) + ' de ' + CAST(@anoEmNúmero
AS VARIABLE),
            @anoEmNúmero,
            @mêsEmNúmero,
            @diaEmNúmero,
            @diaDeSemanaEmNúmero,
            @SemestreEmNúmero,
            @nomeDeSemestre,
            @trimestreEmNúmero,
            @nomeDeTrimestre,
            @anoBissexto,
            @nomeDoMês,
            @abreviaturaDoMês,
            @nomeDeDia,
            @abreviaturaDoDia,
            @séculoEmNúmero,
            @séculoEmNumeraçãoRomana,
            @decadaEmNúmero,
            @decadaPorExtenso,
            @númeroDaSemanaNoAno,
            @numeroDaSemanaDoMês,
            @estaçãoDoAno,
            @finalDeSemana,
            @feriado,
            @nomeDoFeriado,
            @diaUtil,
            @sqlData;
-----Inserir os dados da data, quando Somente sabemos o ANo e Mês -----

        DECLARE @MesTemp VARCHAR(100);
        IF @mêsEmNúmero<10
            SET @MesTemp = '0'+CAST(@mêsEmNúmero AS VARCHAR)
        ELSE
            SET @MesTemp = CAST(@mêsEmNúmero AS VARCHAR)
        IF @diaEmNúmero=1
            INSERT INTO [data]
            SELECT
                CAST(@anoEmNúmero AS VARCHAR)+'-'+@MesTemp+'-00',
                'Sabe-se que é antes de, ' + CAST(@nomeDoMês
AS VARCHAR) + ' de ' + CAST(@anoEmNúmero AS VARCHAR),
                'Sabe-se que é antes de, ' + CAST(@abreviaturaDoMês
AS VARCHAR) + ' de ' + CAST(@anoEmNúmero AS VARCHAR),
                @anoEmNúmero,
                @mêsEmNúmero,null,null,
                @SemestreEmNúmero,
                @nomeDeSemestre,
                @trimestreEmNúmero,
                @nomeDeTrimestre,
                @anoBissexto,
                @nomeDoMês,
                @abreviaturaDoMês,null,null,
                @séculoEmNúmero,
                @séculoEmNumeraçãoRomana,
                @decadaEmNúmero,
                @decadaPorExtenso,
                @númeroDaSemanaNoAno,null,null,null,null,
                null,null,null;

```

```

-----Inserir os dados da data, quando Somente sabemos o ANo -----
IF @diaEmNúmero=1 AND @mêsEmNúmero = 1
    INSERT INTO [data]
        SELECT
            CAST(@anoEmNúmero AS VARCHAR)+'-00-00',
            'Sabe-se que é antes do Ano, '+ CAST(@anoEmNúmero
AS VARCHAR),
            'Sabe-se que é antes do Ano, '+ CAST(@anoEmNúmero
AS VARCHAR),
            @anoEmNúmero,null,null,null,
            null,null,null,null,
            @anoBissexto,null,null,
            null,null,null,
            null,null,null,null,null,null,
            null,null,null,null,null;

        SET @dataInicial = DATEADD(DAY,1,@dataInicial);
    END
END;
GO
EXEC [inserir_dados_data] '1500/01/01', '1900/12/31';

```

## B. Script de criação da dimensão Data (My-SQL)

```
DROP PROCEDURE IF EXISTS `insert data in dimension data`;
DROP TABLE IF EXISTS `dimension data`;

CREATE TABLE `dimension data` (
  `Código da data` INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
  `Data` VARCHAR (50),
  `Descrição completa da data` LONGTEXT,
  `Descrição abreviada da data` LONGTEXT,
  `Ano em número` SMALLINT,
  `Mês em número` SMALLINT,
  `Dia em número` SMALLINT,
  `Dia de Semana em número` SMALLINT,
  `Semestre em número` SMALLINT,
  `Nome de semestre` VARCHAR(20),
  `Trimestre em número` SMALLINT,
  `Nome de trimestre` VARCHAR(20),
  `Ano bissexto` VARCHAR(20) ,
  `Nome do mês` VARCHAR(15),
  `Abreviatura do mês` CHAR(3),
  `Nome de dia` VARCHAR(20),
  `Abreviatura do dia` CHAR(3),
  `Século em número` INT,
  `Século em numeração romana` VARCHAR(8),
  `Década em Número` CHAR(2),
  `Década por Extenso` VARCHAR(50),
  `Número da semana no ano` INT,
  `Numero da semana do mês` INT,
  `Estação do ano` VARCHAR(10),
  `Final de semana` LONGTEXT,
  `Feriado` VARCHAR(30),
  `Nome do Feriado` VARCHAR(50),
  `Dia útil` LONGTEXT,
  `SQL-data` DATE
);

DELIMITER $$

CREATE PROCEDURE `insert data in dimension data`(dataInicial DATE, dataFinal DATE)
BEGIN
  DECLARE Xdate VARCHAR(50);
  DECLARE Xdata DATE;
  DECLARE descriçãoCompletaDaData VARCHAR(100);
  DECLARE descriçãoAbreviadaDaData VARCHAR(100);
  DECLARE anoEmNúmero SMALLINT;
  DECLARE mêsEmNúmero SMALLINT;
  DECLARE diaEmNúmero SMALLINT;
  DECLARE diaDeSemanaEmNúmero SMALLINT;
  DECLARE SemestreEmNúmero SMALLINT;
  DECLARE nomeDeSemestre VARCHAR(20);
  DECLARE trimestreEmNúmero SMALLINT;
  DECLARE nomeDeTrimestre VARCHAR(20);
  DECLARE anoBissexto VARCHAR(20);
  DECLARE nomeDoMês VARCHAR(15);
  DECLARE abreviaturaDoMês CHAR(3);
  DECLARE nomeDeDia VARCHAR(20);
  DECLARE abreviaturaDoDia CHAR(3);
  DECLARE séculoEmNúmero INT;
  DECLARE decadaEmNúmero CHAR(2);
  DECLARE decadaPorExtenso VARCHAR(50);
  DECLARE númeroDaSemanaNoAno INT;
  DECLARE numeroDaSemanaDoMês INT;
  DECLARE séculoEmNumeraçãoRomana VARCHAR(8);
  DECLARE estaçãoDoAno VARCHAR(10);
  DECLARE finalDeSemana VARCHAR(200);
  DECLARE feriado VARCHAR(30);
  DECLARE nomeDoFeriado VARCHAR(50);
  DECLARE diaUtil VARCHAR(20);
  DECLARE SqlData DATE;
```

```

DECLARE MesTemp VARCHAR(100);
DECLARE temp VARCHAR(1);

WHILE dataInicial <= dataFinal
    DO
        SET Xdata = dataInicial;
        SET Xdate = CAST(dataInicial AS CHAR(100));
        SET anoEmNúmero = YEAR(Xdata);
        SET mêsEmNúmero = MONTH(Xdata);
        SET diaEmNúmero = DAY(Xdata);
        SET diaDeSemanaEmNúmero = DAYOFWEEK(Xdata);

        SET trimestreEmNúmero = CASE
            WHEN mêsEmNúmero IN (1,2,3) THEN 1
            WHEN mêsEmNúmero IN (4,5,6) THEN 2
            WHEN mêsEmNúmero IN (7,8,9) THEN 3
            ELSE 4 END;

        SET nomeDeTrimestre = CASE
            WHEN trimestreEmNúmero = 1 THEN 'Primeiro Trimestre'
            WHEN trimestreEmNúmero = 2 THEN 'Segundo Trimestre'
            WHEN trimestreEmNúmero = 3 THEN 'Terceiro Trimestre'
            ELSE 'Quarto Trimestre' END;

        SET SemestreEmNúmero = CASE
            WHEN trimestreEmNúmero IN (1,2) THEN 1
            ELSE 2 END;

        SET nomeDeSemestre = CASE
            WHEN SemestreEmNúmero = 1 THEN 'Primeiro Semestre'
            ELSE 'Segundo Semestre' END;

        SET anoBissexto = CASE
            WHEN (anoEmNúmero % 4) = 0 THEN 'É Ano Bissexto'
            ELSE 'Não é ano Bissexto' END;

        SET nomeDoMês = CASE WHEN mêsEmNúmero = 1 THEN 'Janeiro'
            WHEN mêsEmNúmero = 2 THEN 'Fevereiro'
            WHEN mêsEmNúmero = 3 THEN 'Março'
            WHEN mêsEmNúmero = 4 THEN 'Abril'
            WHEN mêsEmNúmero = 5 THEN 'Maio'
            WHEN mêsEmNúmero = 6 THEN 'Junho'
            WHEN mêsEmNúmero = 7 THEN 'Julho'
            WHEN mêsEmNúmero = 8 THEN 'Agosto'
            WHEN mêsEmNúmero = 9 THEN 'Setembro'
            WHEN mêsEmNúmero = 10 THEN 'Outubro'
            WHEN mêsEmNúmero = 11 THEN 'Novembro'
            ELSE 'Dezembro' END;

        SET abreviaturaDoMês = CASE
            WHEN mêsEmNúmero = 1 THEN 'Jan'
            WHEN mêsEmNúmero = 2 THEN 'Fev'
            WHEN mêsEmNúmero = 3 THEN 'Mar'
            WHEN mêsEmNúmero = 4 THEN 'Abr'
            WHEN mêsEmNúmero = 5 THEN 'Mai'
            WHEN mêsEmNúmero = 6 THEN 'Jun'
            WHEN mêsEmNúmero = 7 THEN 'Jul'
            WHEN mêsEmNúmero = 8 THEN 'Ago'
            WHEN mêsEmNúmero = 9 THEN 'Set'
            WHEN mêsEmNúmero = 10 THEN 'Out'
            WHEN mêsEmNúmero = 11 THEN 'Nov'
            ELSE 'Dez' END;

        SET nomeDeDia = CASE
            WHEN diaDeSemanaEmNúmero = 1 THEN 'Domingo'
            WHEN diaDeSemanaEmNúmero = 2 THEN 'Segunda-Feira'
            WHEN diaDeSemanaEmNúmero = 3 THEN 'Terça-Feira'
            WHEN diaDeSemanaEmNúmero = 4 THEN 'Quarta-Feira'
            WHEN diaDeSemanaEmNúmero = 5 THEN 'Quinta-Feira'
            WHEN diaDeSemanaEmNúmero = 6 THEN 'Sexta-Feira'

```

```

ELSE 'Sábado' END;

SET abreviaturaDoDia = CASE
    WHEN diaDeSemanaEmNúmero = 1 THEN 'Dom'
    WHEN diaDeSemanaEmNúmero = 2 THEN 'Seg'
    WHEN diaDeSemanaEmNúmero = 3 THEN 'Ter'
    WHEN diaDeSemanaEmNúmero = 4 THEN 'Qua'
    WHEN diaDeSemanaEmNúmero = 5 THEN 'Qui'
    WHEN diaDeSemanaEmNúmero = 6 THEN 'Sex'
    ELSE 'Sab' END;

SET séculoEmNúmero = (anoEmNúmero / 100)+1;

SET séculoEmNumeraçãoRomana = CASE
    WHEN séculoEmNúmero = 1 THEN 'I'
    WHEN séculoEmNúmero = 2 THEN 'II'
    WHEN séculoEmNúmero = 3 THEN 'III'
    WHEN séculoEmNúmero = 4 THEN 'IV'
    WHEN séculoEmNúmero = 5 THEN 'V'
    WHEN séculoEmNúmero = 6 THEN 'VI'
    WHEN séculoEmNúmero = 7 THEN 'VII'
    WHEN séculoEmNúmero = 8 THEN 'VIII'
    WHEN séculoEmNúmero = 9 THEN 'IX'
    WHEN séculoEmNúmero = 10 THEN 'X'
    WHEN séculoEmNúmero = 11 THEN 'XI'
    WHEN séculoEmNúmero = 12 THEN 'XII'
    WHEN séculoEmNúmero = 13 THEN 'XIII'
    WHEN séculoEmNúmero = 14 THEN 'XIV'
    WHEN séculoEmNúmero = 15 THEN 'XV'
    WHEN séculoEmNúmero = 16 THEN 'XVI'
    WHEN séculoEmNúmero = 17 THEN 'XVII'
    WHEN séculoEmNúmero = 18 THEN 'XVIII'
    WHEN séculoEmNúmero = 19 THEN 'XIX'
    WHEN séculoEmNúmero = 20 THEN 'XX'
    ELSE 'XXI' END;

SET temp = SUBSTRING(CAST(anoEmNúmero AS CHAR(4)),3,1);
SET decadaEmNúmero= CONCAT(temp,'0');
SET decadaPorExtenso = CONCAT('Década de ', decadaEmNúmero , '
do Século ' ,CAST(séculoEmNumeraçãoRomana AS CHAR(10)));

SET númeroDaSemanaNoAno = WEEKOFYEAR(Xdata);

SET numeroDaSemanaDoMês = CASE
    WHEN diaEmNúmero < 8 THEN 1
    WHEN diaEmNúmero < 15 THEN 2
    WHEN diaEmNúmero < 22 THEN 3
    WHEN diaEmNúmero < 29 THEN 4
    ELSE 5 END;

IF Xdata BETWEEN CONCAT(anoEmNúmero,'/09/23') AND
CONCAT(anoEmNúmero,'/12/20') THEN
    SET estaçãoDoAno = 'Outono';
ELSEIF Xdata BETWEEN CONCAT(anoEmNúmero,'/03/21') AND
CONCAT(anoEmNúmero,'/06/20') THEN
    SET estaçãoDoAno = 'Primavera';
ELSEIF Xdata BETWEEN CONCAT(anoEmNúmero,'/06/21') AND
CONCAT(anoEmNúmero,'/09/22') THEN
    SET estaçãoDoAno = 'Verão';
ELSE
    SET estaçãoDoAno = 'Inverno';
END IF;

IF diaDeSemanaEmNúmero IN(1,7) THEN
    SET finalDeSemana = 'É Final de Semana';
ELSE
    SET finalDeSemana = 'Não É Final de Semana';
END IF;

IF((mêsEmNúmero = 1 AND diaEmNúmero = 1)

```

```

        OR (mêsEmNúmero = 2 AND diaEmNúmero = 28)
        OR (mêsEmNúmero = 5 AND diaEmNúmero = 1)
        OR (mêsEmNúmero = 6 AND diaEmNúmero = 10)
        OR (mêsEmNúmero = 11 AND diaEmNúmero = 1)
        OR (mêsEmNúmero = 12 AND diaEmNúmero = 25)) THEN
            SET feriado = 'É Feriado';
ELSE
    SET feriado = 'Não é Feriado';
END IF;

IF (mêsEmNúmero = 1 AND diaEmNúmero = 1) THEN
    SET nomeDoFeriado = 'Ano Novo';
ELSEIF (mêsEmNúmero = 2 AND diaEmNúmero = 28) THEN
    SET nomeDoFeriado = 'Carnaval';
ELSEIF (mêsEmNúmero = 5 AND diaEmNúmero = 1) THEN
    SET nomeDoFeriado = 'Dia do Trabalhador';
ELSEIF (mêsEmNúmero = 6 AND diaEmNúmero = 10) THEN
    SET nomeDoFeriado = 'Dia de Portugal';
ELSEIF (mêsEmNúmero = 11 AND diaEmNúmero = 1) THEN
    SET nomeDoFeriado = 'Dia de Todos os Santos';
ELSEIF (mêsEmNúmero = 12 AND diaEmNúmero = 25) THEN
    SET nomeDoFeriado = 'Natal';
ELSE
    SET nomeDoFeriado = NULL;
END IF;

IF finalDeSemana = 'É Final de Semana' OR feriado = 'É Feriado'
THEN
    SET diaUtil = 'Não é dia Útil';
ELSE
    SET diaUtil = 'É dia Útil';
END IF;

SET SqlData = CONVERT(Xdata,DATE);

SET descriçãoCompletaDaData = CONCAT(nomeDeDia , ' , ',
diaEmNúmero, ' de ' , nomeDoMês, ' de ' , anoEmNúmero);
SET descriçãoAbreviadaDaData = CONCAT(abreviaturaDoDia , ' , ',
'diaEmNúmero, ' de ' , abreviaturaDoMês, ' de ' , anoEmNúmero);

INSERT INTO `dimension data`(`Data`,`Descrição completa da
data`,`Descrição abreviada da data`,`Ano em número`,`
Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,`
Trimestre em número`,`Nome de trimestre`,`Ano
bissexto`,`Nome do mês`,`Abreviatura do mês`,`
Nome de dia`,`Abreviatura do dia`,`Século em
número`,`Século em numeração romana`,`
Década em Número`,`Década por Extenso`,`Número da semana
no ano`,`Numero da semana do mês`,`Estação do ano`,`
Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)
VALUES (Xdate,
descriçãoCompletaDaData,
descriçãoAbreviadaDaData,
anoEmNúmero,
mêsEmNúmero,
diaEmNúmero,
diaDeSemanaEmNúmero,
SemestreEmNúmero,
nomeDeSemestre,
trimestreEmNúmero,
nomeDeTrimestre,
anoBissexto,
nomeDoMês,
abreviaturaDoMês,
nomeDeDia,
abreviaturaDoDia,
séculoEmNúmero,
séculoEmNumeraçãoRomana,

```

```

decadaEmNúmero,
decadaPorExtenso,
númeroDaSemanaNoAno,
numeroDaSemanaDoMês,
estaçãoDoAno,
finalDeSemana,
feriado,
nomeDoFeriado,
diaUtil,
SqlData);

```

```

INSERT INTO `dimension data`(`Data`,`Descrição completa da
data`,`Descrição abreviada da data`,`Ano em número`,
`Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,
`Trimestre em número`,`Nome de trimestre`,`Ano
bissexto`,`Nome do mês`,`Abreviatura do mês`,
`Nome de dia`,`Abreviatura do dia`,`Século em
número`,`Século em numeração romana`,
`Década em Número`,`Década por Extenso`,`Número da semana
no ano`,`Numero da semana do mês`,`Estação do ano`,
`Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)

```

```

VALUES (Xdate,
CONCAT('Sabe-se que é antes de, ', diaEmNúmero, ' de '
,nomeDoMês, ' de ', anoEmNúmero),
CONCAT('Sabe-se que é antes de, ', diaEmNúmero, ' de '
,abreviaturaDoMês, ' de ', anoEmNúmero),
anoEmNúmero,
mêsEmNúmero,
diaEmNúmero,
diaDeSemanaEmNúmero,
SemestreEmNúmero,
nomeDeSemestre,
trimestreEmNúmero,
nomeDeTrimestre,
anoBissexto,
nomeDoMês,
abreviaturaDoMês,
nomeDeDia,
abreviaturaDoDia,
séculoEmNúmero,
séculoEmNumeraçãoRomana,
decadaEmNúmero,
decadaPorExtenso,
númeroDaSemanaNoAno,
numeroDaSemanaDoMês,
estaçãoDoAno,
finalDeSemana,
feriado,
nomeDoFeriado,
diaUtil,
SqlData);

```

```

INSERT INTO `dimension data`(`Data`,`Descrição completa da
data`,`Descrição abreviada da data`,`Ano em número`,
`Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,
`Trimestre em número`,`Nome de trimestre`,`Ano
bissexto`,`Nome do mês`,`Abreviatura do mês`,
`Nome de dia`,`Abreviatura do dia`,`Século em
número`,`Século em numeração romana`,
`Década em Número`,`Década por Extenso`,`Número da semana
no ano`,`Numero da semana do mês`,`Estação do ano`,
`Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)

```

```

VALUES (Xdate,
CONCAT('Sabe-se que é depois de, ', diaEmNúmero, ' de '
,nomeDoMês, ' de ', anoEmNúmero),

```

```

,abreviaturaDoMês, ' de ' ,anoEmNúmero),
CONCAT('Sabe-se que é depois de, ' ,diaEmNúmero, ' de '
anoEmNúmero,
mêsEmNúmero,
diaEmNúmero,
diaDeSemanaEmNúmero,
SemestreEmNúmero,
nomeDeSemestre,
trimestreEmNúmero,
nomeDeTrimestre,
anoBissexto,
nomeDoMês,
abreviaturaDoMês,
nomeDeDia,
abreviaturaDoDia,
séculoEmNúmero,
séculoEmNumeraçãoRomana,
decadaEmNúmero,
decadaPorExtenso,
númeroDaSemanaNoAno,
numeroDaSemanaDoMês,
estaçãoDoAno,
finalDeSemana,
feriado,
nomeDoFeriado,
diaUtil,
SqlData);

```

```

INSERT INTO `dimension data`(`Data`,`Descrição completa da
data`,`Descrição abreviada da data`,`Ano em número`,
`Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,
`Trimestre em número`,`Nome de trimestre`,`Ano
bissexto`,`Nome do mês`,`Abreviatura do mês`,
`Nome de dia`,`Abreviatura do dia`,`Século em
número`,`Século em numeração romana`,
`Década em Número`,`Década por Extenso`,`Número da semana
no ano`,`Numero da semana do mês`,`Estação do ano`,
`Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)

```

```

VALUES (Xdate,
CONCAT('Sabe-se que é por volta de, ' ,diaEmNúmero, ' de
',nomeDoMês, ' de ' ,anoEmNúmero),
CONCAT('Sabe-se que é por volta de, ' ,diaEmNúmero, ' de
',abreviaturaDoMês, ' de ' ,anoEmNúmero),
anoEmNúmero,
mêsEmNúmero,
diaEmNúmero,
diaDeSemanaEmNúmero,
SemestreEmNúmero,
nomeDeSemestre,
trimestreEmNúmero,
nomeDeTrimestre,
anoBissexto,
nomeDoMês,
abreviaturaDoMês,
nomeDeDia,
abreviaturaDoDia,
séculoEmNúmero,
séculoEmNumeraçãoRomana,
decadaEmNúmero,
decadaPorExtenso,
númeroDaSemanaNoAno,
numeroDaSemanaDoMês,
estaçãoDoAno,
finalDeSemana,
feriado,
nomeDoFeriado,
diaUtil,
SqlData);

```

```

IF mêsEmNúmero<10 THEN
    SET MesTemp = CONCAT('0',CAST(mêsEmNúmero AS CHAR(2)));
ELSE
    SET MesTemp = CAST(mêsEmNúmero AS CHAR(2));
END IF;
IF diaEmNúmero=1 THEN
    INSERT INTO `dimension data` (`Data`,`Descrição completa
da data`,`Descrição abreviada da data`,`Ano em número`,
    `Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,
    `Trimestre em número`,`Nome de trimestre`,`Ano
bissexto`,`Nome do mês`,`Abreviatura do mês`,
    `Nome de dia`,`Abreviatura do dia`,`Século em
número`,`Século em numeração romana`,
    `Década em Número`,`Década por Extenso`,`Número da
semana no ano`,`Numero da semana do mês`,`Estação do ano`,
    `Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)
        VALUES (CONCAT(CAST(anoEmNúmero AS CHAR(4)),'-',MesTemp,'-
00'),
                CONCAT('Sabe-se que é antes de, ' , nomeDoMês, ' de
' ,anoEmNúmero),
                CONCAT('Sabe-se que é antes de, ' ,
abreviaturaDoMês, ' de ' ,anoEmNúmero),
                anoEmNúmero,
                mêsEmNúmero,NULL,NULL,
                SemestreEmNúmero,
                nomeDeSemestre,
                trimestreEmNúmero,
                nomeDeTrimestre,
                anoBissexto,
                nomeDoMês,
                abreviaturaDoMês,NULL,NULL,
                séculoEmNúmero,
                séculoEmNumeraçãoRomana,
                decadaEmNúmero,
                decadaPorExtenso,
                númeroDaSemanaNoAno,NULL,NULL,NULL,NULL,NULL,NULL);
    INSERT INTO `dimension data` (`Data`,`Descrição completa
da data`,`Descrição abreviada da data`,`Ano em número`,
    `Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,
    `Trimestre em número`,`Nome de trimestre`,`Ano
bissexto`,`Nome do mês`,`Abreviatura do mês`,
    `Nome de dia`,`Abreviatura do dia`,`Século em
número`,`Século em numeração romana`,
    `Década em Número`,`Década por Extenso`,`Número da
semana no ano`,`Numero da semana do mês`,`Estação do ano`,
    `Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)
        VALUES (CONCAT(CAST(anoEmNúmero AS CHAR(4)),'-',MesTemp,'-
00'),
                CONCAT('Sabe-se que é depois de, ' , nomeDoMês, '
de ' ,anoEmNúmero),
                CONCAT('Sabe-se que é depois de, ' ,
abreviaturaDoMês, ' de ' ,anoEmNúmero),
                anoEmNúmero,
                mêsEmNúmero,NULL,NULL,
                SemestreEmNúmero,
                nomeDeSemestre,
                trimestreEmNúmero,
                nomeDeTrimestre,
                anoBissexto,
                nomeDoMês,
                abreviaturaDoMês,NULL,NULL,
                séculoEmNúmero,
                séculoEmNumeraçãoRomana,

```

```

                                decadaEmNúmero,
                                decadaPorExtenso,

númeroDaSemanaNoAno, NULL, NULL, NULL, NULL, NULL, NULL, NULL);

        INSERT INTO `dimension data` (`Data`, `Descrição completa da
data`, `Descrição abreviada da data`, `Ano em número`,
                                `Mês em número`, `Dia em número`, `Dia de Semana em
número`, `Semestre em número`, `Nome de semestre`,
                                `Trimestre em número`, `Nome de trimestre`, `Ano
bissexto`, `Nome do mês`, `Abreviatura do mês`,
                                `Nome de dia`, `Abreviatura do dia`, `Século em
número`, `Século em numeração romana`,
                                `Década em Número`, `Década por Extenso`, `Número da
semana no ano`, `Numero da semana do mês`, `Estação do ano`,
                                `Final de semana`, `Feriado`, `Nome do Feriado`, `Dia
útil`, `SQL-data`)
                                VALUES (CONCAT (CAST (anoEmNúmero AS CHAR(4)), '-', MesTemp, '-
00'),
                                CONCAT('Sabe-se que é por volta de, ', nomeDoMês,
' de ', anoEmNúmero),
                                CONCAT('Sabe-se que é por volta de, ',
abreviaturaDoMês, ' de ', anoEmNúmero),
                                anoEmNúmero,
                                mêsEmNúmero, NULL, NULL,
                                SemestreEmNúmero,
                                nomeDeSemestre,
                                trimestreEmNúmero,
                                nomeDeTrimestre,
                                anoBissexto,
                                nomeDoMês,
                                abreviaturaDoMês, NULL, NULL,
                                séculoEmNúmero,
                                séculoEmNumeraçãoRomana,
                                decadaEmNúmero,
                                decadaPorExtenso,

númeroDaSemanaNoAno, NULL, NULL, NULL, NULL, NULL, NULL, NULL);
        END IF;
        IF diaEmNúmero=1 AND mêsEmNúmero = 1 THEN
                INSERT INTO `dimension data` (`Data`, `Descrição completa
da data`, `Descrição abreviada da data`, `Ano em número`,
                                `Mês em número`, `Dia em número`, `Dia de Semana em
número`, `Semestre em número`, `Nome de semestre`,
                                `Trimestre em número`, `Nome de trimestre`, `Ano
bissexto`, `Nome do mês`, `Abreviatura do mês`,
                                `Nome de dia`, `Abreviatura do dia`, `Século em
número`, `Século em numeração romana`,
                                `Década em Número`, `Década por Extenso`, `Número da
semana no ano`, `Numero da semana do mês`, `Estação do ano`,
                                `Final de semana`, `Feriado`, `Nome do Feriado`, `Dia
útil`, `SQL-data`)
                                VALUES (CONCAT (CAST (anoEmNúmero AS CHAR(4)), '-00-00'),
                                CONCAT('Sabe-se que é antes do Ano, ', anoEmNúmero),
                                CONCAT('Sabe-se que é antes do Ano, ', anoEmNúmero),
                                anoEmNúmero, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
                                anoBissexto, NULL, NULL,
                                NULL, NULL, NULL, NULL, NULL);

                INSERT INTO `dimension data` (`Data`, `Descrição completa
da data`, `Descrição abreviada da data`, `Ano em número`,
                                `Mês em número`, `Dia em número`, `Dia de Semana em
número`, `Semestre em número`, `Nome de semestre`,
                                `Trimestre em número`, `Nome de trimestre`, `Ano
bissexto`, `Nome do mês`, `Abreviatura do mês`,
                                `Nome de dia`, `Abreviatura do dia`, `Século em
número`, `Século em numeração romana`,
                                `Década em Número`, `Década por Extenso`, `Número da
semana no ano`, `Numero da semana do mês`, `Estação do ano`,

```

```

        `Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)
        VALUES (CONCAT (CAST (anoEmNúmero AS CHAR (4)), '-00-00'),
                CONCAT ('Sabe-se que é depois do Ano,
',anoEmNúmero),
                CONCAT ('Sabe-se que é depois do Ano,
',anoEmNúmero),
                anoEmNúmero, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
                anoBissexto, NULL, NULL,
                NULL, NULL, NULL, NULL, NULL);

        INSERT INTO `dimension data` (`Data`,`Descrição completa
da data`,`Descrição abreviada da data`,`Ano em número`,`
        `Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,`
        `Trimestre em número`,`Nome de trimestre`,`Ano
bissexto`,`Nome do mês`,`Abreviatura do mês`,`
        `Nome de dia`,`Abreviatura do dia`,`Século em
número`,`Século em numeração romana`,`
        `Década em Número`,`Década por Extenso`,`Número da
semana no ano`,`Numero da semana do mês`,`Estação do ano`,`
        `Final de semana`,`Feriado`,`Nome do Feriado`,`Dia
útil`,`SQL-data`)
        VALUES (CONCAT (CAST (anoEmNúmero AS CHAR (4)), '-00-00'),
                CONCAT ('Sabe-se que é por volta do Ano,
',anoEmNúmero),
                CONCAT ('Sabe-se que é por volta do Ano,
',anoEmNúmero),
                anoEmNúmero, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
                anoBissexto, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
                NULL, NULL, NULL, NULL, NULL);
        END IF;
        SET dataInicial = DATE_ADD (dataInicial, INTERVAL 1 DAY);
    END WHILE;
    INSERT INTO `dimension data` (`Data`,`Descrição completa da
data`,`Descrição abreviada da data`,`Ano em número`,`
        `Mês em número`,`Dia em número`,`Dia de Semana em
número`,`Semestre em número`,`Nome de semestre`,`
        `Trimestre em número`,`Nome de trimestre`,`Ano bissexto`,`Nome
do mês`,`Abreviatura do mês`,`
        `Nome de dia`,`Abreviatura do dia`,`Século em número`,`Século em
numeração romana`,`
        `Década em Número`,`Década por Extenso`,`Número da
ano`,`Numero da semana do mês`,`Estação do ano`,`
        `Final de semana`,`Feriado`,`Nome do Feriado`,`Dia útil`,`SQL-
data`)
        VALUES ('Em Falta',
                'O dado está em Falta',
                'A data é Desconhecida',
                NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
                NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
                NULL, NULL, NULL, NULL, NULL);
END $$
DELIMITER ;

CALL `insert data in dimension data` ('1500/01/01', '1900/12/30');

```

## C. Procedimento para atualização da tabela de Factos (Comissários)

```
DROP PROCEDURE IF EXISTS update_data_in_comissarios;

ALTER TABLE comissarios
  ADD COLUMN `Código Data Inicial` INT AFTER `Data Inicial`,
  ADD FOREIGN KEY fk_dataIni(`Código Data Inicial`) REFERENCES `dimension
data`(`Código da data`) ON DELETE CASCADE;

ALTER TABLE comissarios
  ADD COLUMN `Código Data Final` INT AFTER `Data Final`,
  ADD FOREIGN KEY fk_dataFin(`Código Data Final`) REFERENCES `dimension
data`(`Código da data`) ON DELETE CASCADE;

DELIMITER $$

CREATE PROCEDURE update_data_in_comissarios()
BEGIN
  DECLARE lenComissario INT;
  DECLARE i INT;
  DECLARE dataInicial VARCHAR(10);
  DECLARE dataFinal VARCHAR(10);
  DECLARE anoInicial INT;
  DECLARE anoFinal INT;
  DECLARE diaInicial INT;
  DECLARE diaFinal INT;
  DECLARE mesInicial INT;
  DECLARE mesFinal INT;
  DECLARE sinalInicial CHAR(1);
  DECLARE sinalFinal CHAR(1);
  DECLARE CodDataInicial INT;
  DECLARE CodDataFinal INT;
  DECLARE datafin VARCHAR(20);
  DECLARE dataIni VARCHAR(20);
  DECLARE mesI VARCHAR(2);
  DECLARE mesF VARCHAR(2);
  DECLARE diaI VARCHAR(2);
  DECLARE diaF VARCHAR(2);
  DECLARE lenI INT;
  DECLARE lenF INT;

  SET i=1;

  SELECT MAX(`Referência`) INTO lenComissario FROM comissarios;

  WHILE i <= lenComissario
    DO
      IF (SELECT (`Código Data Inicial`) FROM comissarios WHERE
Referência=i) IS NULL THEN

          SELECT (`Data Inicial`) INTO dataInicial FROM
comissarios WHERE Referência=i;
          SELECT (`Data Final`) INTO dataFinal FROM comissarios
WHERE Referência=i;

          SET anoInicial = CAST(SUBSTRING(CAST(dataInicial AS
CHAR(10)),1,4) AS SIGNED);
          SET mesI = SUBSTRING(CAST(dataInicial AS CHAR(10)),6,2);
          SET mesInicial = CAST(SUBSTRING(CAST(dataInicial AS
CHAR(10)),6,2) AS SIGNED);
          SET diaInicial = CAST(SUBSTRING(CAST(dataInicial AS
CHAR(10)),9,2) AS SIGNED);
          SET diaI = SUBSTRING(CAST(dataInicial AS CHAR(10)),9,2);
          SET sinalInicial = SUBSTRING(CAST(dataInicial AS
CHAR(10)),5,1);

          IF (sinalInicial = '<' OR sinalInicial = '>' OR
sinalInicial = '-') AND mesInicial = 0 AND diaInicial = 0 THEN
            SET dataIni = CONCAT(anoInicial,'-00-00');
```

```

ELSEIF (sinalInicial = '<' OR sinalInicial = '>' OR
sinalInicial = '-') AND mesInicial != 0 AND diaInicial = 0 THEN
    SET dataIni = CONCAT(anoInicial, '-',mesI, '-00');
ELSEIF (sinalInicial = '<' OR sinalInicial = '>' OR
sinalInicial = '-') AND mesInicial != 0 AND diaInicial != 0 THEN
    SET dataIni = CONCAT(anoInicial, '-',mesI, '-',diaI);
ELSEIF sinalInicial = '=' THEN
    SET dataIni='sim';
ELSE
    SET dataIni='nao';
END IF;

IF dataIni = 'sim' THEN
    SELECT (`Código da data`) INTO CodDataInicial FROM
`dimension data` WHERE `Ano em número`= anoInicial AND
    `Dia em número`= diaInicial AND
    `Mês em número` = mesInicial AND
    `Descrição completa da data` NOT LIKE
'Sabe%';
ELSEIF dataIni = 'nao' THEN
    SELECT (`Código da data`) INTO CodDataInicial FROM
`dimension data` WHERE `Data`= 'Em Falta' ;
ELSEIF sinalInicial = '<' THEN
    SELECT (`Código da data`) INTO CodDataInicial FROM
`dimension data` WHERE `Data`=dataIni
    AND `Descrição completa da data` LIKE 'Sabe-se que
é antes%';
ELSEIF sinalInicial = '>' THEN
    SELECT (`Código da data`) INTO CodDataInicial FROM
`dimension data` WHERE `Data`=dataIni
    AND `Descrição completa da data` LIKE 'Sabe-se que
é depois%';
ELSE
    SELECT (`Código da data`) INTO CodDataInicial FROM
`dimension data` WHERE `Data`=dataIni
    AND `Descrição completa da data` LIKE 'Sabe-se que
é por volta%';
END IF;

SET anoFinal = CAST(SUBSTRING(CAST(dataFinal AS
CHAR(10)),1,4) AS SIGNED);
SET mesFinal = CAST(SUBSTRING(CAST(dataFinal AS
CHAR(10)),6,2) AS SIGNED);
SET mesF = SUBSTRING(CAST(dataFinal AS CHAR(10)),6,2);
SET diaFinal = CAST(SUBSTRING(CAST(dataFinal AS
CHAR(10)),9,2) AS SIGNED);
SET diaF = SUBSTRING(CAST(dataFinal AS CHAR(10)),9,2);
SET sinalFinal = SUBSTRING(CAST(dataFinal AS
CHAR(10)),5,1);

IF (sinalFinal = '<' OR sinalFinal = '>' OR sinalFinal =
'-') AND mesFinal = 0 AND diaFinal = 0 THEN
    SET datafin = CONCAT(anoFinal, '-00-00');
ELSEIF (sinalFinal = '<' OR sinalFinal = '>' OR
sinalFinal = '-') AND mesFinal != 0 AND diaFinal = 0 THEN
    SET datafin = CONCAT(anoFinal, '-',mesF, '-00');
ELSEIF (sinalFinal = '<' OR sinalFinal = '>' OR
sinalFinal = '-') AND mesFinal != 0 AND diaFinal != 0 THEN
    SET datafin= CONCAT(anoFinal, '-',mesF, '-',diaF);
ELSEIF sinalFinal = '=' THEN
    SET datafin='sim';
ELSE
    SET datafin='nao';
END IF;

IF datafin = 'sim' THEN
    SELECT (`Código da data`) INTO CodDataFinal FROM
`dimension data` WHERE `Ano em número`= anoFinal AND
    `Dia em número`= diaFinal AND
    `Mês em número` = mesFinal AND

```

```

                                `Descrição completa da data` NOT LIKE
'Sabe%';
                                ELSEIF datafin = 'nao' THEN
                                SELECT (`Código da data`) INTO CodDataFinal FROM
`dimension data` WHERE `Data`= 'Em Falta' ;
                                ELSEIF sinalFinal = '<' THEN
                                SELECT (`Código da data`) INTO CodDataFinal FROM
`dimension data` WHERE `Data`=datafin
                                AND `Descrição completa da data` LIKE 'Sabe-se que
é antes%';
                                ELSEIF sinalFinal = '>' THEN
                                SELECT (`Código da data`) INTO CodDataFinal FROM
`dimension data` WHERE `Data`=datafin
                                AND `Descrição completa da data` LIKE 'Sabe-se que
é depois%';
                                ELSE
                                SELECT (`Código da data`) INTO CodDataFinal FROM
`dimension data` WHERE `Data`=datafin
                                AND `Descrição completa da data` LIKE 'Sabe-se que
é por volta%';
                                END IF;

                                UPDATE    comissarios SET    `Código Data Final` =
CodDataFinal,
                                `Código Data Inicial` = CodDataInicial WHERE
Referência=i;
                                END IF;
                                SET i = i + 1;
                                END WHILE;
END $$
DELIMITER ;

CALL update_data_in_comissarios;

```

## D. Atualizar Províncias

```
DROP PROCEDURE IF EXISTS atualizar_provincias;

DELIMITER $$
CREATE PROCEDURE atualizar_provincias()
BEGIN
    DECLARE      i INT;
    DECLARE      lenLocalidade INT;

    SET i=1;
    SELECT MAX(Referencia) INTO lenLocalidade FROM localidade;
    WHILE i <= lenLocalidade
        DO
            IF (SELECT Província FROM localidade WHERE Referencia=i AND
Província LIKE 'entre%') IS NOT NULL THEN
                UPDATE `localidade` SET `Província`= 'Entre-Douro-Minho'
WHERE Referencia=i;
            END IF;
            SET i = i + 1;
        END WHILE;
END $$
DELIMITER ;

CALL atualizar_provincias;
```

## E. Transformação das coordenadas geográficas

```
ALTER TABLE localidade DROP COLUMN `Latitude1`;
ALTER TABLE localidade DROP COLUMN `Longitude1`;

ALTER TABLE localidade
  ADD COLUMN `Latitude1` FLOAT AFTER `Latitude`;

ALTER TABLE localidade
  ADD COLUMN `Longitude1` FLOAT AFTER `Longitude`;

DROP PROCEDURE IF EXISTS coodernadas_geograficas;

DELIMITER $$
CREATE PROCEDURE coodernadas_geograficas()
BEGIN
  DECLARE lat VARCHAR(255);
  DECLARE lon VARCHAR(255);
  DECLARE la FLOAT;
  DECLARE lo FLOAT;
  DECLARE i INT;
  DECLARE lenLocalidade INT;

  SET i=1;
  SELECT MAX(Referencia) INTO lenLocalidade FROM localidade;
  WHILE i <= lenLocalidade
  DO
    SELECT Latitude INTO lat FROM localidade WHERE Referencia=i;
    SELECT Longitude INTO lon FROM localidade WHERE Referencia=i;

    UPDATE `localidade` SET `Longitude1`=CAST(lon AS
DECIMAL(11,8)),
      Latitude1=CAST(lat AS DECIMAL(11,8)) WHERE Referencia=i;

    SET i = i + 1;
  END WHILE;
END $$
DELIMITER ;

CALL coodernadas_geograficas;

ALTER TABLE localidade DROP COLUMN `Latitude`;
ALTER TABLE localidade DROP COLUMN `Longitude`;

ALTER TABLE localidade CHANGE Latitude1 Latitude FLOAT;
ALTER TABLE localidade CHANGE Longitude1 Longitude FLOAT;
```

## F. Procedimentos auxiliares de data mart ocupações

```
ALTER TABLE localidade CHANGE Longitudel Longitude FLOAT;

DROP TABLE IF EXISTS bbb;
CREATE TABLE `bbb` (
  `id` INT PRIMARY KEY AUTO_INCREMENT,
  `Código de ocupacoes dos comissarios` INT,
  `Código` INT,
  `Eventos` LONGTEXT,
  `Ocupação` LONGTEXT,
  `Código de Ocupação` INT (11)
);

DROP PROCEDURE IF EXISTS aaaaa;

DELIMITER $$

CREATE PROCEDURE `aaaaa`()
BEGIN
  DECLARE lenOcupacoes INT;
  DECLARE i INT;
  DECLARE codigo INT;
  DECLARE oc VARCHAR(255);
  DECLARE sector VARCHAR(255);
  DECLARE estatuto VARCHAR(255);
  DECLARE tipoOcu VARCHAR(255);

  SET i=1;

  SELECT COUNT(id) INTO lenOcupacoes FROM bbb;
  WHILE i <= lenOcupacoes
    DO

      SELECT (`Ocupação`) INTO oc FROM bbb WHERE id =i;
      SELECT `Cód de Ocupação` INTO codigo FROM `ocupação auxiliar` WHERE
`Etiquetas Ocupações` = oc LIMIT 1;

      IF codigo IS NULL THEN
        SET codigo =1024;
      END IF;

      UPDATE `bbb` SET `Código de Ocupação`= codigo WHERE (id) =i;
      SET codigo = NULL;
      SET i = i + 1;
    END WHILE;
END $$
DELIMITER ;

CALL `aaaaa`;
SELECT * FROM bbb

DROP PROCEDURE IF EXISTS ccccc;

DELIMITER $$

CREATE PROCEDURE `cccc`()
BEGIN
  DECLARE lenOcupacoes INT;
  DECLARE i INT;
  DECLARE codigo INT;
  DECLARE cod INT;

  SET i=1;

  SELECT COUNT(id) INTO lenOcupacoes FROM bbb;
  WHILE i <= lenOcupacoes
    DO
```

```
SELECT (`Código de ocupacoes dos comissarios`) INTO cod FROM bbb WHERE
id =i;
SELECT `Código de Ocupação` INTO codigo FROM bbb WHERE id = i;
IF codigo != 1024 THEN
UPDATE `ocupações de comissarios` SET `Código de Ocupação` =
codigo WHERE `Código de ocupacoes dos comissarios` =cod;
END IF;
SET codigo = NULL;
SET i = i + 1;
END WHILE;
END $$
DELIMITER ;
CALL `cccc` ;
SELECT * FROM bbb
```

## G. Processo ETL em data mart ocupações

```
DROP PROCEDURE IF EXISTS `retoque_ocupacoes_comissarios`;

DROP PROCEDURE IF EXISTS `update ocupações de comissarios`;
DROP TABLE IF EXISTS `ocupações de comissarios`;

DELIMITER $$

CREATE PROCEDURE `retoque_ocupacoes_comissarios`()
BEGIN
    DECLARE lenOcupações INT;
    DECLARE i INT;
    DECLARE evento VARCHAR(255);
    DECLARE temp VARCHAR(255);
    DECLARE ocupa VARCHAR(255);
    DECLARE p INT;
    DECLARE j INT;
    SET i=1;

    SELECT COUNT(*) INTO lenOcupações FROM `ocupações de comissarios`;
    WHILE i <= lenOcupações
        DO
            SELECT (`Ocupação`) INTO evento FROM `ocupações de comissarios` WHERE
`Código de ocupacoes dos comissarios` = i;
            SET p = LOCATE('</div>',evento );
            IF (p>0 ) THEN
                SET temp = SUBSTRING_INDEX(evento,'</div>', -1 );
            END IF;
            SET j = LOCATE('<div>',temp);
            IF (j>0 ) THEN
                SET ocupa = SUBSTRING_INDEX(temp,'<div>', -1 );
            END IF;
            IF (p>0 AND j=0 ) THEN
                UPDATE `ocupações de comissarios` SET `Ocupação` = temp WHERE
`Código de ocupacoes dos comissarios` = i;
            END IF;
            IF (p>0 AND j>0 ) THEN
                UPDATE `ocupações de comissarios` SET `Ocupação` = ocupa WHERE
`Código de ocupacoes dos comissarios` = i;
            END IF;

            SET p =0;
            SET j =0;
            SET i = i + 1;

        END WHILE;
END $$
DELIMITER ;

CREATE TABLE `ocupações de comissarios`(
`Código de ocupacoes dos comissarios` INT PRIMARY KEY AUTO_INCREMENT,
`Código`INT(7),
`Eventos` VARCHAR (255),
`Ocupação` VARCHAR (255)
);
INSERT INTO `ocupações de comissarios`(`Código`,`Eventos`)
SELECT `Referência`,`Evento`
FROM comissarios WHERE Evento LIKE '%<Ocupação>%' OR Evento LIKE
'%<div>&lt;Ocupação&gt;%;'

DELIMITER $$

CREATE PROCEDURE `update ocupações de comissarios`()
BEGIN
    DECLARE lenOcupações INT;
    DECLARE i INT;
    DECLARE evento VARCHAR(255);
    DECLARE ocupacao VARCHAR(255);
```

```

DECLARE      temp VARCHAR(255);
DECLARE      j INT;
DECLARE      k INT;
DECLARE      p INT;
DECLARE      final INT;

SET i=0;

SELECT COUNT(*) INTO lenOcupações FROM `ocupações de comissarios`;
WHILE i <= lenOcupações
DO
    SELECT (`Eventos`) INTO evento FROM `ocupações de comissarios` WHERE
`Código de ocupacoes dos comissarios` = i;
    SET p = LOCATE('<div>&lt;Ocupação&gt;',evento );
    SET k = LOCATE('<Ocupação><div>',evento );
    SET j = LOCATE('<Ocupação>',evento );

    IF (k>0) THEN
        SET temp = SUBSTRING_INDEX(evento, '<Ocupação><div>', -1);
    END IF;
    IF (k>0) THEN
        SET ocupacao = SUBSTRING_INDEX(temp, '</div></Ocupação>', 1);
    END IF;
    IF (j>0 AND k=0) THEN
        SET temp = SUBSTRING_INDEX(evento, '<Ocupação>', -1);
    END IF;
    IF (j>0 AND k=0) THEN
        SET ocupacao = SUBSTRING_INDEX(temp, '</Ocupação>', 1);
    END IF;
    IF (p>0 ) THEN
        SET temp = SUBSTRING_INDEX(evento, '<div>&lt;Ocupação&gt;', -1);
    END IF;
    IF (p>0 ) THEN
        SET ocupacao = SUBSTRING_INDEX(temp, '&lt;/Ocupação&gt;</div>',
1);
    END IF;

    UPDATE `ocupações de comissarios` SET `Ocupação` = ocupacao WHERE
`Código de ocupacoes dos comissarios` = i;

    SET p =0;
    SET j =0;
    SET k =0;
    SET final =0;
    SET i = i + 1;

END WHILE;
CALL `retoque_ocupacoes_comissarios`;
END $$
DELIMITER ;

CALL `update ocupações de comissarios`;

```

## H. Criar data mart ocupação

```
DROP TABLE IF EXISTS `ocupação auxiliar`;
CREATE TABLE `ocupação auxiliar` (
  `cod_ocupa` INT PRIMARY KEY,
  `Etiquetas Ocupações` LONGTEXT,
  `Ocupaca` VARCHAR (255),
  `Tipo de Ocupação` VARCHAR (255),
  `Tipo de Sector` VARCHAR (255),
  `Tipo de estatuto` VARCHAR (255)
);
LOAD DATA LOW_PRIORITY LOCAL INFILE 'C:\\Users\\EnoSi\\Desktop\\es.csv'
REPLACE INTO TABLE `comissarios_teste`.`ocupação auxiliar`
CHARACTER SET utf8 FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n';
SELECT * FROM `ocupação auxiliar` WHERE `Tipo de estatuto` = ' ';
DELETE FROM `ocupação auxiliar` WHERE `cod_ocupa`=0;
DELETE FROM `ocupação auxiliar` WHERE `cod_ocupa`=489;
UPDATE `ocupação auxiliar` SET ocupaca='Juiz dos Feitos da Coroa e Fazenda', `Tipo
de Ocupação`='Governo da Res publica'
, `Tipo de Sector` ='Administração', `Tipo de estatuto`='Não mecânico' WHERE
cod_ocupa =489;
UPDATE `ocupação auxiliar` SET ocupaca='Altareiro', `Tipo de Ocupação`='Artes
mecânicas'
, `Tipo de Sector` ='Transformação', `Tipo de estatuto`='Mecânico' WHERE cod_ocupa
=481;
/*insert into `tipo de ocupacao` values ('NULL');
INSERT INTO `tipo de sector` VALUES ('NULL');
INSERT INTO `tipo de estatuto` VALUES ('NULL');*/
SELECT DISTINCT `Tipo de Ocupação` FROM `ocupação auxiliar`
SELECT * FROM `tipo de ocupacao`
DROP TABLE IF EXISTS `ocupação`;
CREATE TABLE `ocupação` (
  `Id` INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  `Ocupa` VARCHAR (255),
  `Tipo de Ocupação1` VARCHAR (255),
  `Tipo de Sector1` VARCHAR (255),
  `Tipo de Estatuto1` VARCHAR (255),
  FOREIGN KEY fk_tipo_ocupacao(`Tipo de Ocupação1`) REFERENCES `tipo de
ocupacao`(`Ocupação`) ON DELETE CASCADE,
  FOREIGN KEY fk_tipo_sector(`Tipo de Sector1`) REFERENCES `tipo de
sector`(`Sector`) ON DELETE CASCADE,
  FOREIGN KEY fk_tipo_estatuto(`Tipo de Estatuto1`) REFERENCES `tipo de
estatuto`(`Estatuto`) ON DELETE CASCADE
)ENGINE=INNODB DEFAULT CHARSET=utf8;
DROP TABLE IF EXISTS `ocupação`;
CREATE TABLE `ocupação` (
  `Id` INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  `Ocupa` VARCHAR (255),
  `Tipo de Ocupação1` VARCHAR (255),
  `Tipo de Sector1` VARCHAR (255),
  `Tipo de Estatuto1` VARCHAR (255)
)ENGINE=INNODB DEFAULT CHARSET=utf8;
INSERT INTO `ocupação`(`Ocupa`,`Tipo de Ocupação1`,`Tipo de Sector1`,`Tipo de
Estatuto1` )
SELECT DISTINCTROW `Ocupaca`,`Tipo de Ocupação`,`Tipo de Sector`,`Tipo de
Estatuto`
FROM `ocupação auxiliar`;

/*INSERT INTO `ocupação`(`Ocupa`,`Tipo de Ocupação1`,`Tipo de Sector1`,`Tipo de
Estatuto1` )
VALUES('Não tem etiqueta ocupação','null', 'null', 'null');*/

DELETE FROM `ocupação` WHERE `Id`=69;

SELECT * FROM `ocupação`
```

## I. Função para procurar ocupação

```
DROP FUNCTION IF EXISTS cod_ocupacao;

DELIMITER $$

CREATE FUNCTION cod_ocupacao (oc VARCHAR(255), sector VARCHAR(255), tipoOcu
VARCHAR(255), estatuto VARCHAR(255))
RETURNS INT
BEGIN
DECLARE codigo INT;
SELECT `Id` INTO codigo FROM `ocupação` WHERE `Ocupa`=oc AND `Tipo de Sector1`=
sector AND
`Tipo de Estatuto1`= estatuto AND `Tipo de Ocupação1`=tipoOcu;
RETURN codigo;
END $$

DELIMITER ;
/*select cod_ocupacao('Abade','Eclesiástico','Igreja','Não mecânico');
SELECT cod_ocupacao('nulo','Ensino','Estudo','Não mecânico');*/
```

## J. Atualizar ocupação

```
DROP FUNCTION IF EXISTS cod_ocupacao;
DELIMITER $$
CREATE FUNCTION cod_ocupacao (oc VARCHAR(255), sector VARCHAR(255), tipoOcu
VARCHAR(255), estatuto VARCHAR(255))
RETURNS INT
BEGIN
DECLARE codigo INT;
SELECT `Id` INTO codigo FROM `ocupação` WHERE `Ocupa`=oc AND `Tipo de Sector1`=
sector AND
`Tipo de Estatuto1`= estatuto AND `Tipo de Ocupação1`=tipoOcu;
RETURN codigo;
END $$
DELIMITER ;
DROP PROCEDURE IF EXISTS `atualizar ocupações`;
ALTER TABLE `ocupação auxiliar`
DROP COLUMN `Cód de Ocupação`;
ALTER TABLE `ocupação auxiliar`
ADD COLUMN `Cód de Ocupação` VARCHAR(255) AFTER `Ocupaca`;
DELIMITER $$
CREATE PROCEDURE `atualizar ocupações`()
BEGIN
DECLARE lenOcupacoes INT;
DECLARE i INT;
DECLARE codigo INT;
DECLARE oc VARCHAR(255);
DECLARE sector VARCHAR(255);
DECLARE estatuto VARCHAR(255);
DECLARE tipoOcu VARCHAR(255);

SET i=1;

SELECT MAX(`cod_ocupa`) INTO lenOcupacoes FROM `ocupação auxiliar`;
WHILE i <= lenOcupacoes
DO

SELECT (`Ocupaca`) INTO oc FROM `ocupação auxiliar` WHERE (`cod_ocupa`) =i;
SELECT (`Tipo de Sector`) INTO sector FROM `ocupação auxiliar` WHERE (`cod_ocupa`)
=i;
SELECT (`Tipo de estatuto`) INTO estatuto FROM `ocupação auxiliar` WHERE
(`cod_ocupa`) =i;
SELECT (`Tipo de Ocupação`) INTO tipoOcu FROM `ocupação auxiliar` WHERE
(`cod_ocupa`) =i;

SET codigo = cod_ocupacao(oc,sector,tipoOcu,estatuto);
UPDATE `ocupação auxiliar` SET `Cód de Ocupação`= codigo WHERE (`cod_ocupa`) =i;

SET i = i + 1;
END WHILE;
END $$
DELIMITER ;

CALL `atualizar ocupações`;

SELECT COUNT(*) FROM comissarios WHERE `Código da Data Inicial` IS NULL;
```

## K. Atualizar ocupação em comissários

```
DROP PROCEDURE IF EXISTS `atualizar ocupações comissarios`;
ALTER TABLE `ocupações de comissarios`
DROP COLUMN `Código de Ocupação`;

ALTER TABLE `ocupações de comissarios`
ADD COLUMN `Código de Ocupação` INT AFTER `Ocupação`;

DELIMITER $$

CREATE PROCEDURE `atualizar ocupações comissarios`()
BEGIN
DECLARE lenOcupacoes INT;
DECLARE i INT;
DECLARE codigo INT;
DECLARE etiquetas VARCHAR(255);
DECLARE cargo VARCHAR(255);
SET i=0;
SELECT MAX(`Código de ocupacoes dos comissarios`) INTO lenOcupacoes FROM `ocupações de comissarios`;
WHILE i <= lenOcupacoes
DO
SELECT (`Ocupação`) INTO cargo FROM `ocupações de comissarios` WHERE `Código de ocupacoes dos comissarios` =i;
SELECT (`Cód de Ocupação`) INTO codigo FROM `ocupação auxiliar` WHERE `Etiquetas Ocupações`=cargo LIMIT 1 ;
IF codigo IS NULL THEN
SELECT (`Id`) INTO codigo FROM `ocupação` WHERE Ocupa ='Não tem etiqueta ocupação';
END IF;

/*call printf(cargo);
call printf(codigo);*/
UPDATE `ocupações de comissarios` SET `Código de Ocupação` = codigo WHERE `Código de ocupacoes dos comissarios` =i;
SET codigo = NULL;
SET i = i + 1;
END WHILE;
END $$
DELIMITER ;

CALL `atualizar ocupações comissarios`;
```

## L. Efetuar ligação e atualização de ocupação na tabela de factos

```
DROP PROCEDURE IF EXISTS `update ocupação na tabela facto`;
ALTER TABLE `comissarios`
DROP COLUMN `Ocupação`;

ALTER TABLE ocupação
CHANGE id `Código de Ocupação` INT;
ALTER TABLE ocupação
CHANGE `Tipo de Ocupação1` `Tipo de Ocupação` VARCHAR(255),
CHANGE `Tipo de Sector1` `Tipo de Sector` VARCHAR(255),
CHANGE `Tipo de Estatuto1` `Tipo de Estatuto` VARCHAR(255);
ALTER TABLE ocupação
CHANGE `Ocupa` `Ocupação` VARCHAR(255);

ALTER TABLE `comissarios`
ADD COLUMN `Código de Ocupação` INT AFTER `Evento`,
ADD FOREIGN KEY fk_ocup(`Código de Ocupação`) REFERENCES `ocupação`(`Código de
Ocupação`) ON DELETE CASCADE;

DELIMITER $$

CREATE PROCEDURE `update ocupação na tabela facto`()
BEGIN
DECLARE lenOcupacoes INT;
DECLARE i INT;
DECLARE idc INT;
DECLARE          codigoOcu INT;

SET i=1;
SELECT MAX(`Código de ocupacoes dos comissarios`) INTO lenOcupacoes FROM `ocupações
de comissarios`;
WHILE i <= lenOcupacoes
DO
SELECT (`Código`) INTO idc FROM `ocupações de comissarios` WHERE `Código de
ocupacoes dos comissarios` =i;
SELECT (`Código de Ocupação`) INTO codigoOcu FROM `ocupações de comissarios` WHERE
`Código de ocupacoes dos comissarios` =i;

UPDATE `comissarios` SET `Código de Ocupação` = codigoOcu WHERE `Referência` = idc;
SET codigoOcu = NULL;
SET idc = NULL;

SET i = i + 1;
END WHILE;
END $$
DELIMITER ;

CALL `update ocupação na tabela facto`;
```

## M. Acrescentar novas ocupações

```
SELECT * FROM ocupação
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (791,'Adobeiro', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (792,'Agente', 'Milícia', 'Defesa', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (793,'Agente', 'Mercancia', 'Negócio', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (794,'Algebrista', 'Artes Mécânicas', 'Saúde', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (795,'Algibebe', 'Artes Mécânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (796,'Bergante', 'Não se sabe', 'Não se sabe', 'Não se sabe');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (797,'Capinha', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (798,'Carameleiro', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (799,'Carcereiro', 'Governo da res publica', 'Justiça', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (800,'Carmelita', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (801,'Cartorário', 'Governo da res publica', 'Administração', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (802,'Criador de cavalos', 'Lavoura', 'Pecuária', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (803,'Eclesiástico', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (804,'Escritor', 'Igreja', 'Não se sabe', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (805,'Provincial', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (806,'Mestre da capela', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (807,'Mestre das obras', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (808,'Mestre de campo', 'Milícia', 'Defesa', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (809,'Mestre da gramática', 'Artes liberais', 'Ensino', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (810,'Mestre de postas', 'Artes mécânicas', 'Transporte', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (811,'Parteira', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação`, `Tipo de Sector`, `Tipo de estatuto`)
VALUES (812,'Partidista', 'Artes liberais', 'Saúde', 'Não mecânico');
```

```

INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (813,'Prateiras', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (814,'Quintador', 'Governo da res publica', 'Administração', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (815,'Terceirão', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (816,'Tutor', 'Governo da casa', 'Administração', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (817,'Vice Reitor', 'Governo da res publica', 'Administração', 'Não
mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (818,'Vive de suas ordens', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (819,'Esparteiro', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (820,'Parteira', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (821,'Carteiro', 'Artes mecânicas', 'Tranporte', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (822,'Esparteiro', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (222,'Chameleiro', 'Artes mecânicas', 'Belas artes', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (823,'Mestre do galeão', 'Artes mecânicas', 'Tranporte', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (824,'Encadeador de contas', 'Artes mecânicas', 'Transformação', 'Mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (825,'Efectuador de casamentos', 'Igreja', 'Eclesiástico', 'Não mecânico');
INSERT INTO ocupação (id,Ocupa, `Tipo de Ocupação1`, `Tipo de Sector1`, `Tipo de
estatuto1`)
VALUES (826,'Efectuador de registos de óbitos', 'Igreja', 'Eclesiástico', 'Não
mecânico');

DELETE FROM ocupação WHERE id = 822

```