

# Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu



# Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu



Dedico este documento à minha família, em especial aos meus avós Alice e Serafim, e à minha tia Ana por todos os sacrifícios que fizeram por mim e pelo seu apoio incondicional.

Dedico também este trabalho à Vanessa Figueiredo, ela acompanhou-me durante todo o meu percurso académico e foi um apoio importante à concretização do meu ciclo de estudos.

Dedico este documento ao meu companheiro Yadi, simplesmente por estar presente e me dar a paz e tranquilidade que precisava para terminar o trabalho.



## **NOTA PRÉVIA**

Esta dissertação deu origem ao artigo: Ferramentas Gratuitas para Desenvolvimento de Soluções de Business Intelligence (Silva, T., Loureiro, J. (2015)). Escola Superior de Tecnologia de Viseu. Enviado e aprovado pela Conferência Anual Portuguesa de Sistemas de Informação 2015.



## RESUMO

O mundo dos negócios é orientado pela tomada de decisão. A diferença entre a decisão certa e a errada é a informação. No mundo corporativo, ganha vantagem quem tem acesso mais rápido às informações que oferecem suporte à gestão empresarial. A *Business Intelligence* (BI) é o processo que recolhe, organiza, analisa, compartilha e monitoriza as informações necessárias para o crescimento de uma organização.

Os dados são armazenados e transformados em informação qualitativa que ajuda a definir as melhores soluções. Isto acontece, pois ao conhecer melhor o negócio e ao ter uma visão sólida, bem fundamentada e completa dos dados corporativos, é possível analisar todos os pontos e arquitectar um planeamento estratégico. No contexto prático existem várias ferramentas que auxiliam os programadores, analistas e responsáveis de negócio a desenvolver aplicações de BI sobre dados existentes em sistemas relacionais (ou outros). No entanto, a maioria está associada a licenciamento. O objectivo do projecto é desenvolver uma plataforma *online* de BI utilizando apenas ferramentas gratuitas, sendo que, para além de um possível produto final, será estudada a viabilidade da(s) ferramenta(s) a utilizar, em projectos futuros.

Neste trabalho são apresentadas várias ferramentas gratuitas que permitem desenvolver aplicações de BI. Existem, actualmente, diversas opções no mercado, desde ferramentas que fazem *reporting*, passando pelo *dashboarding*, *data mining*, entre outras e acabando nas consultas do tipo *Online Analytical Processing* (OLAP). Todas as ferramentas apresentadas foram estudadas e experimentadas. Estas foram alvo de comparação e análise crítica, no fim uma delas será escolhida para um problema de aplicabilidade prática. A ferramenta escolhida foi então utilizada para criar uma prova de conceitos prática, tendo como objetivo mostrar a qualidade da solução produzida por este tipo de ferramenta. Esta destina-se a ser utilizada pela organização Pessoas e Processos, para solucionar um problema real que consiste na possibilidade de obter dashboards e outros elementos BI com ferramentas gratuitas, podendo substituir, em determinadas situações o uso de uma ferramenta paga, com uma qualidade semelhante e assegurando compatibilidade com a infraestrutura informacional pré-existente.

Depois de estudadas várias ferramentas, foi desenvolvida uma aplicação de BI com a prova de conceitos prática. Com este trabalho prova que existem muitas alternativas viáveis para fazer BI de forma gratuita. Dependendo dos objectivos a atingir, as organizações, têm ao dispôr várias ferramentas gratuitas, que podem utilizar para desenvolver soluções de BI. A qualidade das soluções produzidas por estas ferramentas é assinalável, e deve haver um esforço por parte das organizações, em optar por este tipo de *software*. Estas soluções conseguem reduzir os custos das empresas em *software* comercial, que oferecem praticamente os mesmos resultados que as ferramentas gratuitas.



## ABSTRACT

The business world is driven decision making. The difference between the right and the wrong decision is information. In the corporate world, takes advantage those with faster access to information that support business management. The Business Intelligence (BI) is the process that collects, organizes, analyzes, shares and monitors the information necessary for the growth of an organization.

The data is stored and transformed into qualitative information that helps define the best solutions. This is because the best know the business and have a solid vision, well-grounded and full of corporate data, you can analyze all points and at elaborating strategic planning. In practical context there are several tools that help programmers, analysts and business leaders to develop BI applications on existing data in relational systems (or others). However, most are associated with licensing. The goal of the project is to develop an online platform BI using only free tools, and, beyond a possible final product, is studied the feasibility of (s) tool (s) to be used in future projects.

This work presents several free tools that let you develop BI applications. There are now several options on the market provided tools that make reporting, through dashboarding, data mining, among others and ending in consultations type Online Analytical Processing (OLAP). All the tools presented were studied and tried. These were compared to target and critical analysis at the end of which will be chosen for a problem of practical applicability. The chosen tool was then used to create a proof of practical concepts, aiming to show the quality of the solution produced by this type of tool. This is intended to be used by the People and Process organization to solve a real problem which is the ability to get dashboards and other BI elements with free tools and can replace, in certain situations the use of a paid tool with a similar quality and ensuring compatibility with pre-existing informational infrastructure.

After studied various tools, a BI application with proof of practical concepts was developed. This work proves that there are many viable alternatives to BI for free. Depending on the objectives to be achieved, organizations have to dispose of several free tools that can be used to develop BI solutions. The quality of the solutions produced by these tools is remarkable, and there should be an effort by organizations, in opting for this type of software. These solutions can reduce business costs in commercial software, offering virtually the same results as the free tools.



## **PALAVRAS CHAVE**

Business Intelligence

ETL

Reporting

Dashboarding

OLAP

Data Mining

Data Integration

Data Warehouse



## **KEY WORDS**

Business Intelligence

ETL

Reporting

Dashboarding

OLAP

Data Mining

Data Integration

Data Warehouse



## **AGRADECIMENTOS**

A todos os professores que me acompanharam e formaram deixo um agradecimento sincero. Um agradecimento especial ao meu orientador Jorge Loureiro, pela dedicação, ajuda e conselhos que foram vitais para o sucesso deste trabalho.

Agradeço à organização Pessoas e Processos por me ter incentivado a abordar este tema, mas acima de tudo agradeço por me ensinar a ser cada vez melhor profissional e pessoa a cada dia que passa. Agradeço igualmente à organização Martifer, em especial às pessoas José Almeida e Nuno Florentino, pela disponibilidade demonstrada e por me terem incentivado a entrar neste ciclo de estudos.

Agradeço aos meus colegas de trabalho (João Sousa, Júlio Florentino, Roberto Rocha e Tiago Rebelo) as opiniões e críticas dadas durante o desenvolvimento deste trabalho.



# ÍNDICE GERAL

1.	Introdução.....	1
1.1	Objectivos do Trabalho .....	2
1.2	Metodologia de Investigação .....	3
1.3	Estrutura do Documento.....	3
2.	Business Intelligence .....	5
2.1	Enquadramento.....	5
2.2	Arquitectura de um Sistema de BI .....	7
2.3	Vantagens Competitivas de um Projecto de BI.....	8
2.3.1	Business Intelligence – o Processo.....	8
2.4	Return of Investment (ROI): O Valor Acrescentado da BI.....	9
2.4.1	Custos Directos do Projecto .....	10
2.4.2	Custos Indirectos do Projecto.....	11
2.4.3	Medir o Valor do Projecto.....	12
2.5	Aplicações de Business Intelligence .....	12
2.5.1	Data Integration/ETL e Data Quality .....	13
2.5.2	Reporting.....	14
2.5.3	Dashboarding e Scorecarding.....	15
2.5.4	Análise e Cubos OLAP .....	16
2.5.5	Data Mining.....	17
2.5.6	Aplicações Avançadas.....	18
3.	Ferramentas Gratuitas para o Desenvolvimento de Soluções de Business Intelligence .....	21
3.1	Software Livre.....	21
3.2	Ferramentas de ETL.....	23
3.2.1	Talend Open Studio.....	23
3.2.2	Pentaho Data Integration.....	25
3.2.3	Comparação entre o Talend Open Studio e o Pentaho Data Integration .....	26
3.3	Ferramentas de Reporting .....	28
3.3.1	JasperReports .....	28
3.3.2	BIRT Project .....	30
3.3.3	Comparação entre o JasperReports e o BIRT Project .....	31

3.4	Ferramentas de OLAP .....	33
3.4.1	Mondrian.....	33
3.4.2	JPivot .....	35
3.4.3	Comparação entre o Mondrian e o JPivot.....	37
3.5	Ferramentas de Data Mining .....	38
3.5.1	Weka.....	38
3.5.2	Rapid Miner.....	39
3.5.3	Comparação entre o Weka e o Rapid Miner.....	41
3.6	Ferramentas Suite .....	41
3.6.1	TIBCO Jaspersoft .....	42
3.6.2	Actuate BIRT.....	44
3.6.3	Jedox Palo.....	46
3.6.4	Pentaho .....	47
3.6.5	SpagoBI .....	49
3.7	Comparação entre as Ferramentas .....	52
3.7.1	Ferramentas Não Suite.....	52
3.7.2	Ferramentas Suite .....	54
3.8	Ferramenta Escolhida .....	57
4.	Problema para Aplicabilidade Prática.....	61
4.1	Sistema de Controlo e Acompanhamento de Projectos Internos .....	61
4.1.1	Sistema de Informação para Gestão de Contratos .....	62
4.1.2	Sistema de Informação para Controlo e Acompanhamento de Projectos.....	66
4.1.3	Módulo de Gestão de BackOffice.....	68
4.1.4	Módulo de Business Intelligence – Portal do Conhecimento .....	69
5.	Criação do Data Warehouse e dos Processos de ETL .....	75
5.1	Sistema Transaccional .....	75
5.2	Construção do Data Warehouse .....	77
5.3	Processos de ETL .....	82
5.3.1	Seleção do Workspace e do Projecto .....	82
5.3.2	Criação dos Metadados e Ligação ao Sistema Fonte.....	83
5.3.3	Criação dos Processos .....	84
5.3.4	Deployment dos Processos de ETL e Agendamento da Execução.....	91
6.	Criação da Aplicação de Business Intelligence .....	93
6.1	Tecnologias Utilizadas .....	93

6.1.1	ASP MVC .....	93
6.1.2	Javascript.....	94
6.1.3	Estilos Responsivos.....	96
6.2	Resultado Final – Aplicação de BI.....	97
6.2.1	Página de Login.....	97
6.2.2	Organização da Aplicação.....	98
6.2.3	Página de Entrada.....	100
6.2.4	Gráficos e Dashboards .....	101
6.2.5	Reports .....	106
6.2.6	Informação Geográfica.....	109
6.3	Outros Aspectos .....	109
7.	Conclusões e Trabalho Futuro.....	111
	ANEXO I – SISTEMAS DE GESTÃO DE INFORMAÇÃO NAS ORGANIZAÇÕES .....	119
	ANEXO II – BASES DE DADOS E DATA WAREHOUSING.....	125



## ÍNDICE DE FIGURAS

Figura 1 - Comparação entre a popularidade do software comercial e o software open-source .....	2
Figura 2 – Estrutura usual de um sistema de business intelligence.....	7
Figura 3 - Processo de BI .....	9
Figura 4 – Exemplo de uma ferramenta de ETL Kettle .....	13
Figura 5 – Exemplo da ferramenta de data quality Talend Data Quality .....	14
Figura 6 – Exemplo de ferramenta de reporting Crystal Reports.....	15
Figura 7 – Exemplo de dashboard.....	16
Figura 8 – Exemplo de scorecard.....	16
Figura 9 – Exemplo de ferramenta de OLAP JPivot.....	17
Figura 10 – Exemplo de ferramenta de data mining RapidMiner .....	17
Figura 11 – Plataforma de BI na área de e-commerce .....	18
Figura 12 – Plataforma de BI na área de recursos humanos .....	19
Figura 13 - Criação de um processo de ETL no Talend Open Studio.....	24
Figura 14 – Seleccção de projecto e workspace no Talend Open Studio.....	25
Figura 15 - Criação de um processo de ETL no Pentaho Data Integration .....	26
Figura 16 – Criação de um relatório utilizando o Jaspersoft Studio .....	29
Figura 17 – Criação de um relatório utilizando o BIRT Designer .....	31
Figura 18 – Editar cubos de dados na Mondrian Schema Editor .....	34
Figura 19 – Exemplo de utilização da ferramenta JPivot.....	36
Figura 20 – Menu do JPivot .....	36
Figura 21 – Exemplo de consulta feita no JPivot.....	37
Figura 22 – Colaboração entre o Mondrian e o JPivot.....	38
Figura 23 – Exemplo de aplicação de clustering utilizando a ferramenta Weka .....	39
Figura 24 – Exemplo de aplicação de daa mining utilizando o RapidMiner .....	40
Figura 25 – Arquitectura do Jaspersoft – versão completa .....	43
Figura 26 – Exemplo de dashboards criadas com o Jaspersoft – versão completa .....	44
Figura 27 - Exemplo de dashboards criadas com o BIRT iHub.....	45
Figura 28 – Pluggin Palo para Excel a correr no Microsoft Excel.....	47
Figura 29 – Componentes que formam o Pentaho .....	48
Figura 30 – Exemplo de dashboard criada com o Pentaho .....	49
Figura 31 – Componentes que formam o SpagoBI .....	50
Figura 32 – Exemplo de dashboard criado com o SpagoBI .....	52
Figura 33 – Ciclo de vida de um contrato .....	62
Figura 34 – Página de listagens de contratos .....	63
Figura 35 – Página de edição de contrato (1).....	63
Figura 36 – Página de edição de contrato (2).....	64
Figura 37 – Fluxo de vida de uma factura.....	64
Figura 38 – Listagem de facturas .....	65
Figura 39 – Página de edição de factura .....	65
Figura 40 – Contrato com factura associada .....	66
Figura 41 – Fluxo de vida de um projecto .....	66

Figura 42 – Página de edição de projecto.....	67
Figura 43 - Página de edição de projecto (2).....	67
Figura 44 - Página de edição de projecto (3).....	68
Figura 45 – Página de edição de uma entidade de negócio .....	68
Figura 46 – Distribuição do número de projectos por estado.....	70
Figura 47 – Análise financeira plurianual .....	70
Figura 48 – Distribuição geográfica dos projectos .....	71
Figura 49 – Investimento feito por natureza.....	71
Figura 50 – Parâmetros/filtros de pesquisa do dashboard .....	72
Figura 51 – Comparação entre o valor planeado.....	72
Figura 52 – Execução de investimento por programa .....	72
Figura 53 – Detalhe sobre o investimento, execução .....	73
Figura 54 – Tabelas do sistema transaccional sobre os projectos.....	76
Figura 55 – Tabelas do sistema transaccional sobre os projectos (2).....	76
Figura 56 - Tabelas do sistema transaccional sobre os projectos (3).....	77
Figura 57 – Estrutura do DW desenvolvido .....	78
Figura 58 – Dimensões do DW desenvolvido .....	79
Figura 59 - Factos do DW desenvolvido .....	80
Figura 60 – Criação do projecto no workspace por omissão.....	82
Figura 61 - Criação do projecto no workspace por omissão (2).....	83
Figura 62 – Tipos de metadados possíveis de criar no TOS.....	83
Figura 63 – Metadados da ligação à base de dados transaccional .....	84
Figura 64 – Metadados da ligação ao DW .....	84
Figura 65 - Diagrama de actividades com o fluxo de ETL seguido .....	85
Figura 66 – Subprocesso inicial (ETL_MASTER) .....	85
Figura 67 – Subprocesso ETL_START.....	86
Figura 68 – Mapeamento de dados entre fonte origem e destino, ‘ .....	86
Figura 69 – Subprocesso ETL_DIMENSOES .....	87
Figura 70 – Subprocesso que insere dados na dimensão DIM__PROJ_MEDIDA.....	88
Figura 71 – Mapeamento entre a tabela transaccional e a dimensão do DW .....	88
Figura 72 – Eliminação dos dados existentes numa tabela.....	89
Figura 73 – Subprocesso ETL_END.....	89
Figura 74 – Mapeamento de dados com atribuição de valor calculado.....	90
Figura 75 – Escolher a opção de actualização dos dados .....	91
Figura 76 – Agendamento do processo de ETL no SpagoBI .....	92
Figura 77 – Padrão MVC .....	94
Figura 78 – Template responsivo utilizado na aplicação de BI criada .....	96
Figura 79 – Página de login.....	97
Figura 80 – Aplicação com template responsivo.....	98
Figura 81 – Menu da aplicação.....	99
Figura 82 – Menu de topo .....	99
Figura 83 – Corpo de uma página .....	100
Figura 84 – Indicadores numéricos da página de entrada.....	100
Figura 85 – Gráfico com evolução do número de projectos ao longo do tempo.....	101
Figura 86 – Gráfico com distribuição dos projectos por programa .....	101

Figura 87 – Gráficos com filtros do SpagoBI .....	103
Figura 88 - Gráficos com filtros externos .....	103
Figura 89 – Dashboard sem filtros .....	104
Figura 90 - Dashboard com filtros internos.....	105
Figura 91 - Dashboard com filtros externos.....	105
Figura 92 – Relatório com a lista de projectos (formato pdf) .....	107
Figura 93 - Relatório com a lista de projectos (formato excel).....	107
Figura 94 – Relatório embutido na aplicação com parâmetros externos.....	108
Figura 95 – Exportação de relatório com parâmetros externos .....	108
Figura 96 – Relatório embutido com parâmetros internos .....	108
Figura 97 – Exemplo de componente com informação geográfica.....	109



## ÍNDICE DE TABELAS

Tabela 1 - Análise comparativa entre o Talend Open Studio e o Pentaho Data Integration .....	27
Tabela 2 - Análise comparativa entre o JasperReports e o BIRT Project .....	31
Tabela 3 - Análise comparativa entre o Weka e o Rapid Miner .....	41
Tabela 4 – Comparação entre as funcionalidades disponibilizadas .....	55
Tabela 5 – Comparação entre os sistemas operativos suportados .....	56
Tabela 6 - Comparação entre as versões disponíveis .....	56
Tabela 7 – Comparação quanto à curva de aprendizagem .....	58
Tabela 8 – Comparação do número e qualidade de documentação existente .....	59
Tabela 9 – Comparação do suporte técnico.....	59



## ABREVIATURAS E SIGLAS

BI	Business Intelligence
OLAP	Online Analytical Processing
PHP	Hypertext Preprocessor
DW	Data Warehouse
ETL	Extract, Transform and Load
SWOT	Strengths, Weaknesses, Opportunities and Threats
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
SCM	Supply Chain Management
ROI	Return of Investment
KPI	Key Performance Indicator
FSP	Free Software Foundation
OSI	Open Source Initiative
LGPL	Lesser General Public License
TOS	Talend Open Studio
IDE	Integrated Development Environment
PDI	Pentaho Data Integration
XML	eXtensible Markup Language
SGBD	Sistemas Gestores de Bases de Dados
EPL	Eclipse Public License
MDX	Multidimensional Expressions
JOLAP	Java Online Analytical Processing
JDBC	Java Database Connectivity
API	Application Programming Interface
JSP	Java Server Pages
CPL	Commercial Pilot Licence
GPL	General Public License
AGPL	Affero General Public License
YALE	Yet Another Learning Environment
PUC	Pentaho User Console
PAC	Pentaho Administration User
QBE	Query by Example
SCD	Slowing Changing Dimensions
SK	Surrogate Keys
URL	Uniform Resource Locator



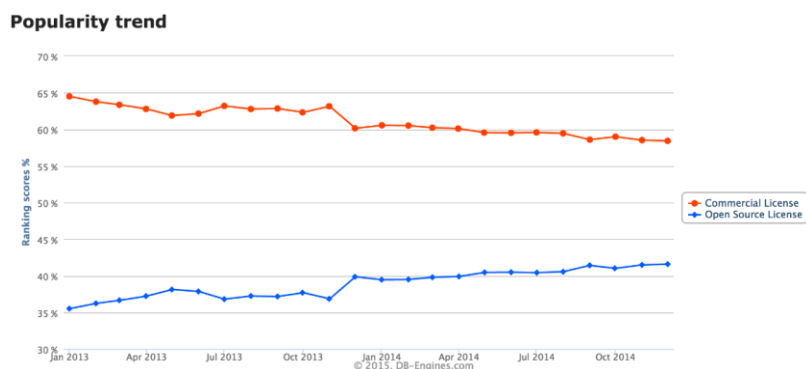
# 1. Introdução

Os factores de instabilidade económica, que se têm revelado nos últimos anos [1][2], levam a que as empresas façam cada vez mais esforços para reduzirem as despesas e os custos inerentes às actividades praticadas [5][8]. Um dos maiores problemas actuais das organizações é a incapacidade de investimento na optimização dos processos de negócio. As empresas não se comprometem a estes investimentos, pois falta-lhes capacidade de investimento e optimização [9]. Para se perceber melhor a importância dos sistemas de informação nas organizações ver o Anexo 1.

Perante este contexto, existe a necessidade de oferecer soluções de inovação e optimização dos processos de negócio das organizações [8]. A *Business Intelligence* (BI) [3][4] surge para ajudar a resolver este problema [10]. O conceito de BI já tem alguns anos, mas apenas recentemente (século XXI) tem ganho popularidade e aceitação por parte das empresas. A utilização de práticas de BI exige que existam dados de actividades operacionais. Depois, estes devem ser transformados em informação útil, e esta informação deve gerar conhecimento suficiente para apoiar a tomada de decisão por parte dos gestores de negócio. Os dados provenientes das actividades empresariais servem de matéria-prima para as ferramentas de BI. Estas, por sua vez, têm como principal objectivo a integração e cruzamento de dados, exibindo os mesmos sob forma gráfica [2]. As ferramentas de BI permitem, a partir do cruzamento de informação, extrair conhecimento relevante (p.e: na área das vendas, identificar as categoria de produtos mais vendidos numa localização geográfica) e apontar problemas (p.e: na área da construção civil, identificar projectos atrasados face ao planeamento inicial) [6]. Os dois problemas anteriores permitem aos gestores de negócio tomar decisões importantes, e este é o grande objectivo das ferramentas de BI [3]. Quer sejam decisões para otimizar, corrigir ou solidificar, o essencial é que estas sejam importantes para o negócio onde são aplicadas.

Devido ao facto da maioria das ferramentas de BI terem um custo elevado [7], muitas vezes não comportável para as empresas devido ao clima económico actual já referido anteriormente, o foco deste trabalho é investigar ferramentas de BI gratuitas para uso comercial. As ferramentas gratuitas e/ou *open source* são implementadas e mantidas, na sua maioria, por comunidades de utilizadores (que

não têm objectivos lucrativos no processo) ou equipas internas de empresas com outros produtos comercializáveis. A utilização de *software* gratuito pode solucionar o problema de investimento tecnológico das empresas, existindo já muitas ferramentas que apresentam uma alternativa viável e com qualidade comparativamente com outras opções comerciais [11] (p.e: Open Office, Hypertext Preprocessor (PHP), MySQL, Notepad++, etc.). Na Figura 1 podemos constatar que a popularidade de *software* gratuito (neste caso *open-source*) tem ganho terreno face ao *software* comercial.



**Figura 1 - Comparação entre a popularidade do *software* comercial e o *software* open-source**  
(fonte: <http://www.wired.com/2015/01/open-source-database/>)

O foco deste trabalho serão as ferramentas gratuitas para fazer BI. Será realizada uma pesquisa, tendente à sua identificação, descrição, experimentação e análise crítica das mesmas. Para mostrar a viabilidade deste tipo de soluções será abordado um caso prático, e para o resolver será utilizada pelo menos uma ferramenta resultante do levantamento anterior. A implementação prática tem por objectivo sustentar que as tecnologias de BI gratuitas são uma solução alternativa viável e exequível, permitindo inovação e optimização das actividades e processos de negócios das organizações. No entanto, desengane-se quem pensar que não existem qualquer tipo de custos, pois não é o facto das ferramentas serem gratuitas que significa que não existam custos com outros componentes [12] de um projecto de BI, mas estes custos (os directos e indirectos) serão discutidos mais à frente neste documento.

### 1.1 Objectivos do Trabalho

O grande objectivo deste trabalho é aprender mais sobre *business intelligence*, e como se pode fazer esta através de ferramentas gratuitas. Identifica-se um problema que é a necessidade de melhorar e optimizar os processos de negócio das organizações através de BI feita com ferramentas com custos reduzidos. As principais contribuições são:

- Procurar métodos para a redução dos custos tecnológicos das empresas;
- Salientar a importância da *Business Intelligence*;
- Utilizar ferramentas gratuitas como solução para o problema de investimento tecnológico;
- Empreender uma análise crítica e comparativa entre diferentes ferramentas de BI;
- Apresentar as características que ajudam à escolha das ferramentas a escolher;
- Referir e descrever soluções para os processos de extracção de dados e implementação de um projecto de BI.

Os objectivos anteriores são de máxima importância, mas o principal objectivo deste trabalho é mostrar que as ferramentas gratuitas para o desenvolvimento de soluções de BI são uma alternativa viável às ferramentas comerciais, e que conseguem competir com estas na qualidade dos resultados apresentados.

Pretende-se também que este documento seja o mais simples e directo possível, mas não descuidando os enquadramentos teóricos da *business intelligence*. O conteúdo deste documento não é estritamente teórico. Depois de apresentados os elementos teóricos, serão apresentadas as ferramentas estudadas e será feita uma análise crítica e comparativa das mesmas. Por fim será apresentado um caso real para a aplicação da(s) ferramenta(s) com mais potencial, e pretende-se, com a demonstração prática, atingir o objectivo principal deste trabalho.

### 1.2 Metodologia de Investigação

Será necessário fazer um estudo do estado da arte das ferramentas de BI existentes no mercado que sejam gratuitas. Para isso será feita uma investigação de maneira a fazer um levantamento das opções disponíveis [13][14]. Após a fase de pesquisa das ferramentas disponíveis, será preciso fazer uma avaliação experimental das mesmas, de maneira a estudar a sua viabilidade. O trabalho foi desenvolvido para a organização Pessoas e Processos (<https://www.pessoaseprocessos.com>), e juntamente com a equipa de BI desta, consideram-se viáveis aquelas que contenham um conjunto de módulos que:

- Possibilitem extrair dados de várias fontes;
- Permitam transformar e carregar os dados para diferentes fontes de dados;
- Permitam fazer *reporting* e *dashboarding*;
- Sejam de fácil manuseamento.

Serão experimentadas diversas ferramentas. No caso de duas ou mais satisfazerem os requisitos anteriores, serão escolhidas aquelas que apresentarem um tempo de desenvolvimento menor. Após a selecção da(s) ferramenta(s) a utilizar é necessário fazer um levantamento da informação a apresentar e a forma como vai ser apresentada.

Após a escolha da(s) ferramenta(s) de BI a utilizar e do levantamento dos dados e forma como se devem apresentar é que se entra na fase de execução do projecto.

### 1.3 Estrutura do Documento

O documento encontra-se estruturado em várias secções. Estas apresentarão os fundamentos teóricos, o estudo realizado e a aplicabilidade prática dos conhecimentos adquiridos.

A secção 2, Business Intelligence, apresentará todos os conceitos teóricos sobre este tema. O objectivo desta secção é apresentar o conceito, assim como a importância da BI nas organizações, apresentando as suas vantagens competitivas, o processo e o retorno de investimento das soluções de BI. Nesta

secção serão ainda apresentadas as formas como se pode fazer BI, desde os *reportings*, passando pelas *dashboards* e acabando nas consultas OLAP, data mining e aplicações avançadas.

A secção 3, Ferramentas Gratuitas para o Desenvolvimento de Soluções de Business Intelligence, apresentará o estudo realizado sobre as soluções existentes actualmente para fazer BI com ferramentas gratuitas. Nesta secção serão apresentadas várias ferramentas estudadas, desde as ferramentas individuais (que respondem a uma área específica da BI) até às ferramentas integradas (que contêm um conjunto de módulos que lidam com áreas diferentes da BI). As ferramentas apresentadas serão alvo de comparação e análise crítica, e no fim da secção, de acordo com requisitos definidos pela organização, será escolhida a ferramenta, que para este trabalho, se considera mais promissora.

Na secção 4, Problema para Aplicabilidade Prática, serão apresentadas as aplicações já existentes na organização. O sistema transaccional será apresentado nesta secção. Este tem os dados, que devem ser mostrados na aplicação de BI criada neste trabalho. Com a apresentação dos sistemas, o transaccional e a aplicação de BI (ambos já existentes e criados pela organização), o leitor conseguirá perceber o porquê da existência destes sistemas, assim como dos objectivos dos mesmos.

Na secção 5, Criação do Data Warehouse e dos Processos de ETL, será apresentado o Data Warehouse (DW) desenvolvido no âmbito deste trabalho. Antes de apresentar o DW criado, no âmbito deste trabalho, será apresentada a base de dados do sistema transaccional. Esta apresentação tem como objectivo dar a conhecer algumas tabelas e campos importantes neste sistema. Posteriormente, como se apresentará nesta secção, estas tabelas e campos serão a fonte dos dados a inserir no DW criado. Depois de apresentado o sistema fonte, será apresentada a arquitectura do DW escolhida, assim como tabelas que o compõem. Por último, serão apresentados os processos de *Extract, Transform and Load* (ETL). Estes serão os responsáveis por extrair a informação, do sistema fonte, e inserir a mesma no DW.

Na secção 6, Criação da Aplicação de Business Intelligence, será apresentada a aplicação de BI criada no contexto deste trabalho. As tecnologias usadas para a criação da aplicação serão apresentadas nesta secção. Os componentes criados no SpagoBI serão também apresentados nesta secção.

Por fim, na secção 7, serão apresentadas as conclusões finais deste trabalho. Business Intelligence

## 2. Business Intelligence

### 2.1 Enquadramento

*Business Intelligence* é a designação algo generalista e ampla estando actualmente relacionada com um conjunto de processos de negócio, *software* e ferramentas tecnológicas específicas [2][3]. Os seus grandes objectivos sempre foram: recolher dados, transformar os mesmos em informação, e, iterativamente, tornar esta útil e crucial para a tomada de decisões de negócio. Podemos recuar no tempo para obter um paralelismo para a BI. Para isto teremos de nos lembrar da evolução da arte da estratégia.

*Intelligence* [15] significa a posse de informação privilegiada que conduz os seus mestres a terem vantagem competitiva sobre os demais rivais, sendo vital para uma estratégia de sucesso. Sun Tzu (*A Arte da Guerra* [16], há 2500 anos) escreveu as primeiras palavras sobre a noção de estratégia. Nos séculos XV-XVI seguiu-se Maquiavel [17]. Estas obras são a base para outras abordagens actuais, no que toca à gestão empresarial. Exemplo disso mesmo é a análise *Strengths, Weaknesses, Opportunities and Threats* (SWOT [18]). Em síntese, num contexto onde existe concorrência e competição, ter conhecimento é, inevitavelmente, uma grande vantagem.

Com a chegada dos primeiros computadores [19] (as dantescas *mainframes* [20] da década de 1960), as primeiras tarefas de armazenamento e automatização tiveram o seu início. Contudo, o poder de processamento era bastante baixo, havia falta de estruturas de conectividade para a transmissão e consequente partilha de dados e a verificavam-se, amiúde, incompatibilidades entre sistemas. Isto tudo dificultou a execução de processos organizacionais. *Reports* de dados simples chegavam a demorar semanas ou mais para serem executados. No entanto, nas últimas décadas os esforços para melhorar o processamento de dados fez com que a evolução fosse enorme. Têm-se verificado ao longo dos últimos anos alterações no ambiente de negocio que implicaram:

- Maior exigência na qualidade dos produtos/serviços;
- Conhecimento mais aprofundado de mercados e clientes com maior segmentação;
- Muita concorrência na maior parte dos mercados;
- Racionalização de processos internos e redução de custos;
- Conhecimento, controlo e monitorização dos riscos de negócio.

As bases de dados têm vindo a crescer a um ritmo exponencial. Os sistemas *Enterprise Resource Planning* (ERP) [21][22], *Customer Relationship Management* (CRM) [23][24], *Supply Chain Management* (SCM) [25], *Data Warehouses*[26][27] e Internet oferecerem diversa informação aos decisores de negócio. É então necessário transformar dados em informação e esta em conhecimento. Por último, actualmente é notório que as organizações se transformaram de modo a serem cada vez mais horizontais, menos hierarquizadas e com maior autonomia nas decisões operacionais. Isto requer a divulgação de informação crítica junto de mais colaboradores, entre eles, os gestores. Todos estes acontecimentos implicaram um aumento da vigilância no processo de monitorização interna, na gestão da *performance* e nos sistemas de informação para suporte à decisão.

Os sistemas de *business intelligence* são, actualmente, os “catalisadores” da mudança, permitindo praticar estes novos métodos de “boa gestão” das empresas. Eficazmente, pelo diagnóstico, análise, partilha e *reporting* de dados, os gestores conseguem entender o que é vital nos negócios onde estão envolvidos e conseguem transformar as grandes quantidades de dados em informação e conhecimento. Sendo assim, é natural transformar estes mesmos conhecimentos em resultados, para que as empresas possam passar da eficiência operacional à eficácia corporativa. Existem diversas definições para a *business intelligence*, mas no geral este é um conceito que engloba um vasto conjunto de aplicações de apoio à tomada de decisão que possibilitam um acesso rápido, partilhado e interactivo das informações, bem com a sua análise e manipulação; através destas ferramentas, os utilizadores podem descobrir relações e tendências e transformar grandes quantidades de informação em conhecimento útil [2][3].

Actualmente, existem um conjunto de processos de negócio e actividades, consideradas críticas, no modelo individual da organização, assim como a sua respectiva cadeia de valor [28]. Estes são essenciais para a aquisição de conhecimento essencial para os processos de decisão. O quadro abaixo (quadro 1) mostra algumas realidades relevantes no contexto de BI que são posteriormente materializadas em diversos processos de negócio:

**Quadro 1 – Valor acrescentado da BI e a sua aplicabilidade empresarial**

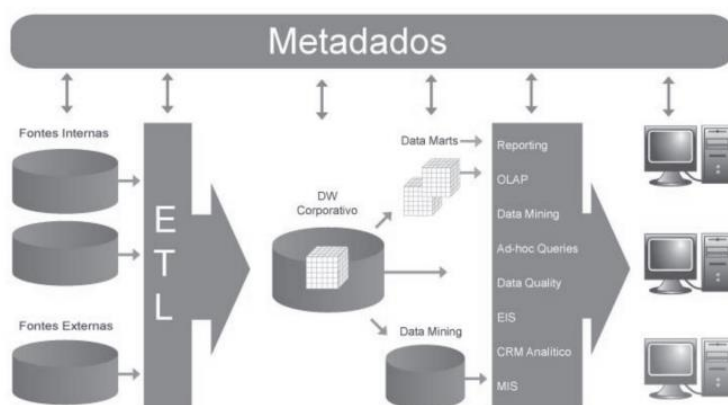
<b>Comercial</b> <ul style="list-style-type: none"><li>– Análise do comportamento do consumidor</li><li>– Análise da rentabilidade de consumidores/segmentos</li><li>– Análise de <i>cross-selling</i></li><li>– Análise da força de vendas</li><li>– Análise dos canais de distribuição</li></ul>	<ul style="list-style-type: none"><li>– Consolidação financeira</li><li>– <i>Reporting</i> financeiro</li></ul>
<b>Marketing</b> <ul style="list-style-type: none"><li>– Penetração no mercado/segmentos</li><li>– Eficácia das campanhas de <i>marketing</i> (análise de meios)</li><li>– Análise do ciclo de vida do produto/serviço</li></ul>	<b>Operações/Logística</b> <ul style="list-style-type: none"><li>– Eficiência operacional</li><li>– Planeamento da produção</li><li>– Controlo de qualidade</li><li>– Análise da cadeia logística</li></ul>
<b>Finanças</b> <ul style="list-style-type: none"><li>– Previsão, planeamento e orçamentação</li><li>– Análise de <i>performance</i></li></ul>	<b>Recursos Humanos</b> <ul style="list-style-type: none"><li>– Planeamento da afectação de recursos</li><li>– Avaliação de <i>performance</i></li><li>– Análise da compensação</li><li>– Avaliação de competências</li></ul>

Numa primeira análise, os sistemas de BI partilham um conjunto de objectivos fundamentais [3]:

- **Fiabilidade nos dados:** a fácil integração entre áreas é muito importante para a consciencialização da gestão, mas o mais importante passa pela fiabilidade dos dados partilhados;
- **Transparência e visão de negócio:** o conhecimento em tempo real (“O quê?”, “Quanto?”, “Quando?”, “Onde?” e “Como?”) faz com que os decisores tenham uma perspectiva das áreas que devem gerir com transparência e aumentar a capacidade de compreensão (o “Porquê?”);
- **Tomada de decisão:** apenas a interpretação correcta do contexto de negócio pode permitir tomadas de decisão eficientes, então, a informação sobre a forma de conhecimento, produzida pelos sistemas de *business intelligence*, com o suporte das tecnologias de informação modernas, tem como dever o suporte e a justificação de tomadas de decisão pelos demais intervenientes na gestão do negócio.

### 2.2 Arquitectura de um Sistema de BI

Os responsáveis por estes sistemas devem lembrar-se que um sistema de BI não existe “isolado”. Este está interligado com as fontes de dados de um ou mais negócios, sejam estas fontes bases de dados transaccionais ou simples ficheiros com dados organizacionais, ou seja, tudo o que se considera uma fonte crítica de informação dos processos de negócio é candidata a fornecer dados ao sistema de *business intelligence*. Perceber a interacção entre a informação existente e os seus consumidores finais, através de ferramentas de análise e visualização, permite entender a importância da informação que foi obtida, filtrada e produzida. A Figura 2 mostra a arquitectura típica de um sistema de BI [2].



**Figura 2 – Estrutura usual de um sistema de *business intelligence***  
(fonte: <http://pt.slideshare.net/thomasdacosta/cost-management-maximizes-business-intelligence-roi>)

Usualmente um sistema de *business intelligence* é composto pelos seguintes componentes [2]:

- **Módulo de ETL:** dedica-se, como o próprio nome indica, à extracção, transformação e carregamento dos dados, tendo como objectivo a extracção da informação de várias fontes (sistemas CRM, sistemas ERP, ficheiros de texto, ficheiros separados por vírgula, ficheiros Excel, etc.);

- **Data Warehouse/Data Marts:** repositório onde são guardados os dados extraídos dos sistemas fonte. Existem diversas vantagens em existir um local físico à parte para guardar os dados a mostrar nas aplicações de BI. A principal é o facto de conseguirmos guardar informação histórica, podendo ser agregada para mostrar em análises complexas;
- **Aplicação de BI:** responsável por mostrar a informação do projecto de BI aos utilizadores finais. A forma de organizar a aplicação depende de problema para problema, mas a utilização de relatórios, análises OLAP ou *dashboards* são métodos bastante utilizados para disponibilizar os dados para análise.

### 2.3 Vantagens Competitivas de um Projecto de BI

Um projecto de BI é um investimento cujo retorno são as vantagens competitivas de gestão da informação e do conhecimento, por parte da organização que o implementa. Deve existir uma visão incremental para encarar e avaliar as vantagens de um projecto deste tipo. O que se pretende, com este tipo de visão, é o desenvolvimento faseado e gradual com o objectivo de dar aos gestores de negócio novas funcionalidades de análise e acrescentar valor à informação na forma de conhecimento, e que permite a tomada de decisões. O objectivo essencial das aplicações de *Business Intelligence* é entregar a informação correcta às pessoas com capacidade de decisão e conhecimento do negócio em causa e dentro do *timing* adequado, isto para que as melhores decisões sejam potenciadas com o melhor *ratio* de performance (custos e benefícios) possível. As suas vantagens competitivas serão reais, se forem atendidas as seguintes premissas [29]:

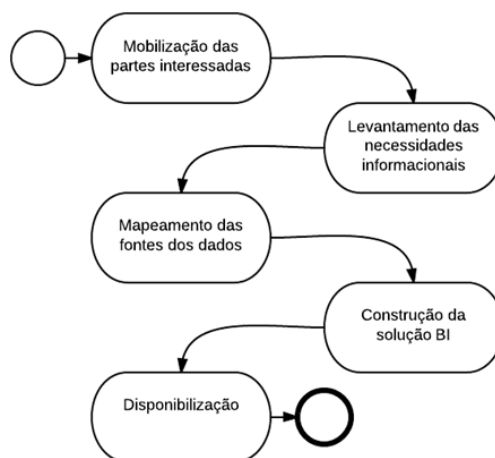
- **Primeira:** é necessária a definição de modelos de dados, inventário, transformação e partilha de informação, sendo obrigatório a coerência do conhecimento originado;
- **Segunda:** devem-se constituir indicadores para optimização da qualidade da informação empresarial. Quando se procura um conhecimento empresarial e multidimensional, os projectos de BI devem melhorar os processos de negócio e originar uma maior exigência na produtividade dos envolvidos;
- **Terceira:** um projecto de BI deve assegurar uma ligação forte entre a visão de negócio e a visão técnica/tecnológica da empresa.

#### 2.3.1 Business Intelligence – o Processo

São várias as actividades necessárias para a identificação, desenvolvimento e implantação de um sistema de BI. Estas actividades estão intimamente ligadas e são imprescindíveis para o êxito da solução de BI.

Para o completo entrosamento, é importante conhecer o processo macro subjacente ao desenvolvimento do sistema de apoio a decisão. O fluxo das actividades fornece esclarecimento e a abstracção do funcionamento da solução de BI. Precisamos compreender cada uma destas actividades, que seguem um ordenamento lógico e de fluxo simples para construção de BI dentro da organização.

O processo, mostrado na Figura 3, é composto de várias etapas e os seus elementos compreendem as actividades de mobilização dos participantes (directos e indirectos), levantamento das necessidades organizacionais, mapeamento da origem dos dados, construção da solução e, por fim, a disponibilização para os consumidores das informações (os utilizadores).



**Figura 3 - Processo de BI**

Segue-se a descrição de cada uma destas actividades [30]:

- **Mobilização das partes interessadas:** para início do projecto de BI, todos os *stakeholders* devem ser mobilizados;
- **Identificação das necessidades informacionais:** na actividade de levantamento das necessidades procura-se entender qual a informação exigida pelos gestores. Nesta etapa, procura-se listar todas as necessidades sem haver preocupação, neste momento, com a existência "real" dos dados;
- **Mapeamento das fontes dos dados:** o mapeamento da origem dos dados é feito após a atividade anterior. Aqui, verifica-se a viabilidade das necessidades efetuadas na etapa anterior, caso exista viabilidade são mapeadas as fontes dos dados para a posterior extracção na construção do sistema de BI;
- **Construção da solução de BI:** nesta etapa é iniciada a construção propriamente dita da solução. Trata-se, sem dúvida alguma, da maior etapa do processo. Todas as actividades de extracção e averiguação da qualidade dos dados são realizadas nesta etapa;
- **Disponibilização:** depois de percorridas todas as etapas anteriores, chega-se à última etapa: a disponibilização. Esta é uma etapa muito delicada, pois trata-se do momento onde é entregue o produto ao utilizador final. Se a etapa de mobilização realizada no início do processo tiver êxito, dificilmente existirão problemas na disponibilização.

As cinco etapas constituem uma visão geral de todo o processo necessário para a completa implementação de uma solução de BI. Também se trata de uma prática recomendada, pois caso sejam desrespeitadas algumas destas etapas, provavelmente levantar-se-ão dificuldades e problemas futuros que poderão ser críticos no sucesso ou continuidade do projecto. Logo, a qualidade final do projecto de BI vai depender muito da atenção dada a cada uma destas actividades.

### 2.4 Return of Investment (ROI): O Valor Acrescentado da BI

Ao longo dos últimos dez anos, a implementação de sistemas de informação cresceram exponencialmente. Isto fez com que a análise custo-benefício fosse cada vez mais exigente para novos projectos. Nestes últimos anos foram vários os desafios impostos aos responsáveis pelas tecnologias

de informação, desde integrações dos processos e operações empresariais (sistemas ERP), a ligação entre a empresa e os seus clientes (sistemas CRM) ou até mesmo um melhor aproveitamento da velocidade de conexão à Internet. Todos estes desafios mostraram que a tecnologia não é um fim nela própria mas sim um meio para concretizar projectos. É importante haver um investimento cuidado quando se usa a tecnologia, caso contrário, o valor acrescentado do projecto diminui. A abordagem de “pensar em grande” dominava o mercado das tecnologias de informação há cerca de dez anos, mas, entretanto, houve uma mudança de paradigma. Agora o que se pretende é que se continue a “pensar em grande” mas começando “por baixo” e com evolução constante. Esta atitude, mais moderada, deve-se à existência de vários projectos/empresas que, depois do grande *boom* inicial, foram sempre descendo no seu número de implementações, qualidade das serviços fornecidos e, como é óbvio, viram muito diminuída a sua carteira negocial. A introdução de tecnologias de informação, e de *Business Intelligence* em particular, tem como objectivo a criação de um ambiente calmo e auto-regulado, onde a ligação entre as pessoas, tecnologias e processos consiga trazer valor à organização. A quantificação deste valor acrescentado subtraindo os diversos investimentos (iniciais e contínuos) serve como cálculo do ROI (*return of investment*– retorno do investimento) [30]. O ROI é um *ratio* que calcula a relação custo-benefício:

$$\text{ROI} = (\text{Lucros para a organização/custos do desenvolvimento}) \times 100$$

O valor do tempo sob uma perspectiva financeira é muito importante, pois este tempo entra no cálculo do ROI.

### 2.4.1 Custos Directos do Projecto

Existem vários tipos de custos quando se está a implementar um projecto de *business intelligence*, que são claros e simples de calcular. Não é apenas o *software* e o *hardware* que devem ser quantificados nos custos. Os custos relativos à sustentabilidade do projecto devem ser também considerados (e.g.: custos de infra-estruturas e formação, tanto dos programadores como dos utilizadores).

Seguidamente são apresentados vários tipos de custos líquidos do projecto, assim como uma breve descrição de cada tipo [31]:

- **Custos de aquisição:** todas as compras de *software* devem ser contabilizadas, desde os custos de *data warehousing* (incluindo as licenças com bases de dados relacionais e/ou multidimensionais, assim como os custos dos servidores, memória, energia, etc.), passando pelas ferramentas para implementar BI (*dashboarding*, *reporting*, OLAP, ETL, etc.) até às aplicações de *data quality*;
- **Custos de TI:** as actualizações de *software* e do *hardware* inicialmente escolhidos ou o aumento da capacidade da rede de comunicação fazem parte dos custos das TI. Todos os custos das infra-estruturas necessárias entram neste tipo de custos e ao longo do tempo, deve haver o cuidado de actualização e monitorização das redes de dados;
- **Custos de desenvolvimento:** os recursos humanos necessários às tarefas para o lançamento, implementação e “disponibilização” do produto ao utilizador final devem ser considerados ao longo da implementação do projecto. O número de pessoas pode variar ao longo do tempo, e todas as entradas e saídas devem ser consideradas na folha de cálculo das despesas com os recursos humanos. O tipo de colaboradores que entram num projecto deste tipo são as pessoas

internas da organização e os consultores externos. Quanto aos colaboradores externos o valor do custo é facilmente quantificado pois os valores são acertados com antecedência. Por outro lado, os colaboradores internos não estão sempre alocados cem por cento ao projecto, então deve haver o cuidado de registar as horas de implementação por parte de cada colaborador;

- **Custos de formação:** a formação é importante tanto para os utilizadores finais como para os *developers* e outros responsáveis de gestão e manutenção. No entanto, a formação não deve ser vista nunca como um “remendo” ao mau trabalho desenvolvido, e quanto mais usável e robusto for o produto desenvolvido menor será o custo da formação necessária aos utilizadores finais. Outros custos como: manuais de apoio, instalações e tempo que os colaboradores internos ocuparam com a formação devem ser também considerados;
- **Custos de manutenção:** depois do produto ser lançado são inevitáveis as actualizações nas funcionalidades desenvolvidas (tanto correctivas como evolutivas), assim como os custos de manutenção da infra-estrutura. No entanto, não se deve confundir manutenção com implementação e novas funcionalidades, sempre que o âmbito de uma funcionalidade mude drasticamente é necessária a capacidade de negociação de modo a que a manutenção se transforme num novo desenvolvimento. Os custos de manutenção envolvem, usualmente, a atribuição dos mesmos pelos outros tipos de custos.

### 2.4.2 Custos Indirectos do Projecto

Os custos indirectos são, fundamentalmente, os custos da compatibilidade entre as ferramentas e os custos da sua utilização [31][2]. Nos dias que correm é usual as organizações terem mais do que uma ferramenta de BI diferente, cada uma delas responsável por funcionalidades específicas. Por vezes até existe mais do que uma ferramenta para fazer o mesmo, isto porque podem existir requisitos específicos de negócio que são respondidos por uma ferramenta em particular. Estes custos indirectos devem ser adicionados à equação do cálculo do ROI, pois o tempo que estas “manutenções” requerem pode inflacionar o tempo gasto.

Através das ferramentas analíticas e de *reporting* que são introduzidas num projecto de *Business Intelligence*, é esperado que sejam detectadas e avaliadas evoluções quantitativas nos processos de negócio [3], através das melhores tomadas de decisão, e por vezes até se verifica a melhoria em alguns processos organizacionais através da aliança entre os demais decisores de negócio. No entanto, por vezes, existem erosões no resultado do projecto de BI, e estas devem ser consideradas nos custos indirectos. Um dos problemas deve-se à baixa aceitação por parte dos utilizadores finais. Estes, com alguma frequência, revelam alguma resistência à mudança, que se deve essencialmente à falta de conhecimento ou formação, ou, e este é o principal motivo de resistência, à falta de existência de mudanças nos sistemas, processos de negócio e relação de gestão da informação onde o processo de BI é implementado. A falta de integração e trabalho em equipa eficaz entre os utilizadores e os gestores de negócio é também um dos problemas de compromisso verificados nas organizações. Estes problemas podem verificar-se noutros tipos de projecto, por exemplo, quando existe a construção de um novo centro comercial. Se depois de planificado e criado, não se conseguir atingir um número mínimo de clientes, não pode, ser considerado um sucesso, por mais que arquitecturalmente ou tecnologicamente seja bem-sucedido. Na prática, deve haver sempre a consciencialização do contexto da organização onde se insere o projecto de BI.

### 2.4.3 Medir o Valor do Projecto

A avaliação dos custos de um projecto de BI é bastante simples, apenas havendo alguma dificuldade ao auferir os custos indirectos que vimos anteriormente, mas o cálculo do valor dos seus benefícios pode revelar-se mais complexo. A relação entre a aquisição de valor e o custo das ferramentas de BI nem sempre é calculado de forma fácil. Existem, principalmente, dois constrangimentos [32]:

- As ferramentas tecnológicas não são auto-suficientes. Antes de funcionarem é necessário que sejam programadas por nós. O valor criado pelas ferramentas depende de muitos factores dentro da organização, por vezes bastante diferentes, dependendo do nível hierárquico;
- As soluções de BI são, a cada dia que passa, mais integradas com os restantes sistemas organizacionais, sendo que estes não são constantes ao longo do tempo, logo as sucessivas mudanças e *updates* são difíceis de imputar a um ou mais componentes específicos.

Mesmo com estes constrangimentos, é vital haver um esforço de gestão e tentar fazer uma simulação dos valores adquiridos ao longo de um intervalo temporal de modo a haver uma quantificação do valor do projecto, nem que seja apenas uma aproximação do valor real.

Vamos fazer um exercício simples: imaginemos um projecto de BI que tem como objectivo a disponibilização de uma plataforma de BI *web-based*. Esta plataforma mostra os dados de vendas de N produtos ao longo do tempo em geografias diferentes, estes dados são mostrados através de várias *dashboards* e *reports*. O principal objectivo passa por aumentar as vendas de uma categoria de produtos numa escala global. Imaginemos o seguinte:

- **Vendas da categoria *x* antes da nova plataforma de BI:** 500.000 euros
- **Vendas da categoria *x* depois de aplicadas decisões de negócio com base na nova plataforma de BI:** 830.000 euros

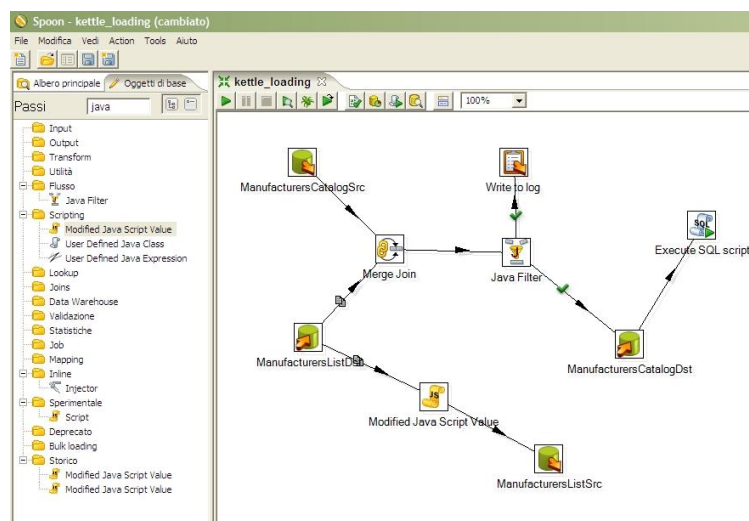
Independentemente do tempo de desenvolvimento e manutenção, se as restantes variáveis se mantiverem constantes, concluímos que o valor em benefícios de negócio ronda os 330.000 euros. Se o custo do projecto (custos directos e custos indirectos) fosse na ordem dos 55.000 euros, então teríamos um ROI de 600% ((Benefícios / Custos)\*100).

## 2.5 Aplicações de Business Intelligence

Existem várias formas de implementar BI, mas independentemente da forma com se faz, o importante são sempre os dados, pois o que varia na forma de fazer BI é essencialmente a forma como se apresentam os dados. Já foi referido que em BI os dados são mostrados numa forma qualitativa de modo a que os responsáveis de negócio consigam [2], através dos dados, obter informação crucial acerca do negócio, ou seja, independentemente da forma como se apresentam os dados devem-se apresentar numa forma qualitativa, resumida e fácil para o utilizador final conseguir tirar conclusões que auxiliem a tomada de decisões.

### 2.5.1 Data Integration/ETL e Data Quality

O processo de ETL [33][34][35] é o responsável pela extracção, transformação e carregamento de dados das mais variadas fontes para o DW. A utilização de ferramentas que tratam da integração de dados diminuiu o tempo de transporte de dados. Estas costumam ter *interfaces* à base de *drag'n'drop* de componentes que tratam de diversos aspectos (desde a extracção até ao carregamento), sendo cada vez mais comum a parametrização dos mesmos componentes, ou de todo o processo de ETL, através de código fonte (p.e: java, c#, c++, etc.). A Figura 4 mostra um exemplo de *interface* de uma ferramenta de ETL. Nesta consegue-se perceber que a integração de dados é feita usando vários componentes. Estes componentes têm as mais diversas funções, alguns tratam da extracção de dados, outros fazem mapeamento de dados entre fontes diferentes, etc.



**Figura 4 – Exemplo de uma ferramenta de ETL Kettle**  
(fonte: [http://www.robertomarchetto.com/talend\\_studio\\_vs\\_kettle\\_pentao\\_pdi\\_comparison](http://www.robertomarchetto.com/talend_studio_vs_kettle_pentao_pdi_comparison))

Os processos de ETL foram evoluindo ao longo dos anos e chegou a um ponto em que o objectivo já não é apenas fazer integração de dados mas também manter a qualidade dos mesmos. Surgem então ferramentas ETL com módulos que tratam da *data quality* [36][37]. O grande objectivo da *data quality* é assegurar que os dados cumprem os requisitos mínimos de qualidade e coerência. Por norma, um processo de *data quality* é muito parecido com um processo de ETL ou é usado neste (na etapa da transformação de dados). A Figura 5 mostra um exemplo de aplicação de *data quality*. Nesta consegue-se perceber que, tal como as aplicações de integração de dados, são usados vários componentes. Estes componentes, neste tipo de ferramentas, têm funções como: verificar a integridade de dados numa fonte seleccionada, identificar dados que não estejam de acordo com regras de negócio, correcção automática de dados (por exemplo, corrigir um código postal de um endereço de um país através da localidade), etc.

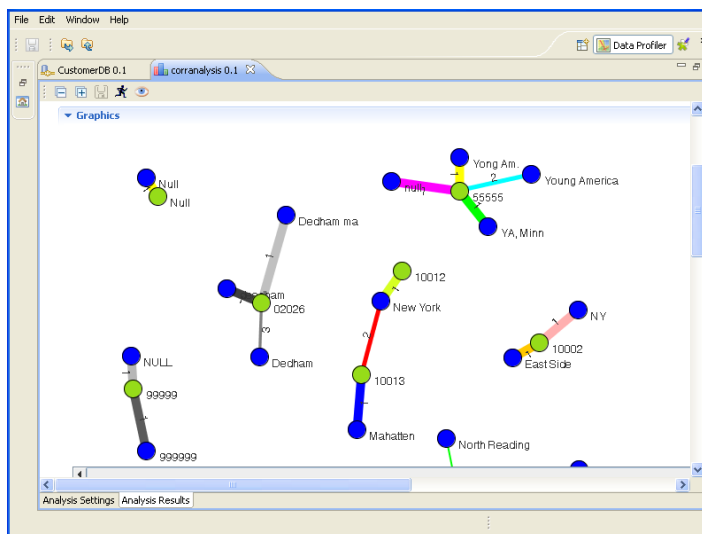


Figura 5 – Exemplo da ferramenta de *data quality* Talend Data Quality  
(fonte: [http://www.robertomarchetto.com/talend\\_studio\\_vs\\_kettle\\_pentao\\_pdi\\_comparison](http://www.robertomarchetto.com/talend_studio_vs_kettle_pentao_pdi_comparison))

### 2.5.2 Reporting

O objectivo dos *reports* [38][39] é acompanhar as tarefas de uma empresa ao longo do tempo. Em certas situações, este *reporting* é feito diariamente. Os dados que aparecem num *report* dizem respeito a um processo de negócio e podem chegar aos utilizadores finais através de vários formatos. Os mais comuns são: pdf, word e excel. Existem *reports*, que em certos casos, podem ser parametrizados de maneira a que os utilizadores possam filtrar a informação final que aparecerá no documento ou *interface*, são por isso *reports* não estáticos. Existe outro tipo de *report* cada vez mais utilizado que é o *ad-hoc report*. Este, para lá de disponibilizar filtros aos utilizadores, permite que estes possam ainda escolher como e o que deverá aparecer no *report* final. Independentemente do tipo de *reporting* utilizado, as principais questões que devem ser levadas em consideração são [40]:

- Avaliar as melhores fontes de dados e identificar os elementos que têm que aparecer num *report*;
- Determinar o período de tempo em que devem ser disponibilizados os *reports*;
- Identificar quais as ferramentas e formatos que devem ser utilizados;
- Identificar os utilizadores e grupo de utilizadores dos *reports* finais.

A Figura 6 mostra um exemplo de aplicação para desenvolver *reporting*.

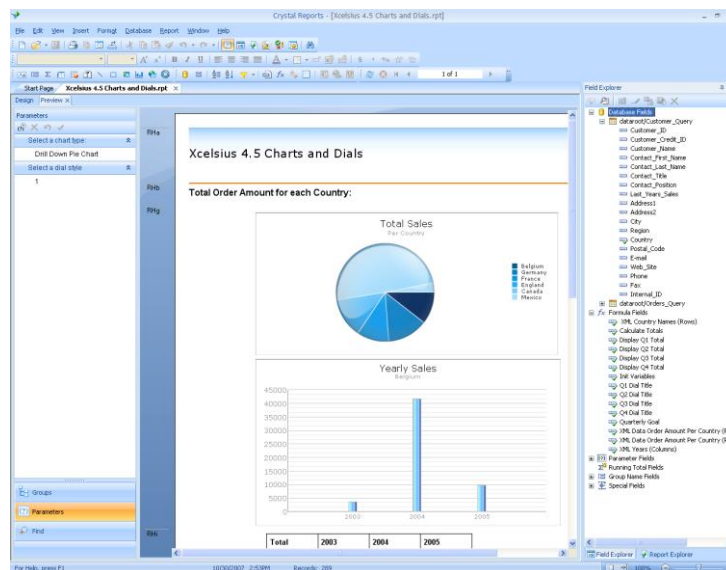


Figura 6 – Exemplo de ferramenta de *reporting* Crystal Reports  
(fonte: [http://en.wikipedia.org/wiki/Crystal\\_Reports](http://en.wikipedia.org/wiki/Crystal_Reports))

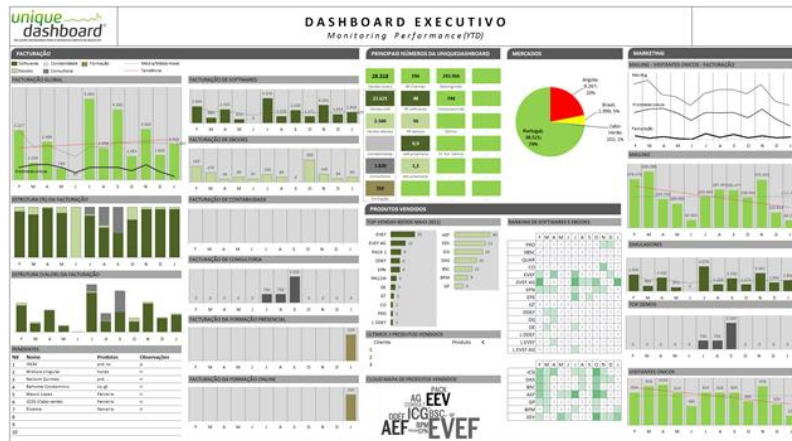
### 2.5.3 Dashboarding e Scorecarding

O significado de *dashboard* pode ser traduzido como painel de informação [41][42]. É essencialmente um instrumento de gestão para a monitorização que tem por objectivo servir os decisores de topo, intermédios ou colaboradores, na visualização e análise da informação crítica (e.g.: *Key Performance Indicators* (KPI [43])), de modo a sustentar a tomada de decisão.

O *Dashboard* apresenta um conjunto de especificidades próprias, que caracterizam actualmente o seu conceito [42]:

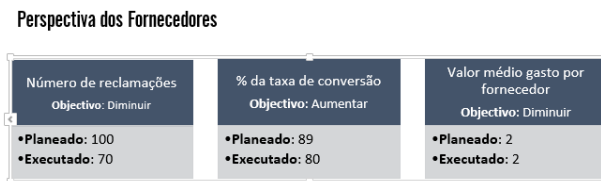
- É um essencialmente um instrumento de apoio à decisão;
- Expõe rapidamente os principais indicadores de uma organização, área, unidade orgânica, projecto, etc.;
- Tenta apresentar informação num único ecrã/página A4;
- Possui uma apresentação gráfica simples, objectiva mas também elegante;
- Utiliza técnicas de design de modo a reduzir o "lixo visual" e para dar maior eficácia à transmissão da informação/mensagem;
- Tenta combinar diferentes variáveis sobre diferentes perspectivas de modo a expor relações que seriam difíceis de identificar analisando os mesmos dados em separado;
- Deve permitir a interacção entre o analista e os dados (exemplo: customizar, segmentar, seleccionar, aprofundar, etc.).

A Figura 7 mostra como a informação pode ser mostrada sob o formato de *dashboard*. A *dashboard* da figura, apresenta vários componentes que apresentam os dados de forma sumária em gráficos e indicadores. Estes componentes conseguem expôr rapidamente os indicadores que ajudam no apoio à decisão. A informação é apresentada, nestes componentes, de forma simples e elegante, aumentando assim a eficácia da informação que se pretende apresentar.



**Figura 7 – Exemplo de dashboard**  
(fonte: [http://www.radardeprojetos.com.br/2015\\_04\\_01\\_archive.html](http://www.radardeprojetos.com.br/2015_04_01_archive.html))

Um *scorecard* [44] é um instrumento que melhora o *reporting* tradicional, pois utiliza metas de avaliação definidas pela organização que o utiliza. O principal objectivo dos *scorecards* é medir o desempenho de cada área de negócio analisada. A principal diferença entre um *dashboard* e um *scorecard* é que enquanto o primeiro visa mostrar informação, o segundo mostra a mesma informação, mas com referência a metas de cumprimento, ou seja, KPI's. Os *scorecards* são muito utilizados em sistemas de avaliação de competências, sistemas financeiros, entre outros. Essencialmente, este instrumento avalia o comportamento actual com aquele que é expectável. A Figura 8 mostra um exemplo de *scorecard*.



**Figura 8 – Exemplo de scorecard**

Como se pode ver pela imagem anterior existe um *reporting* com metas a atingir e objectivos. Este *report*, pelas razões que já foram referidas, é um *scorecard*.

### 2.5.4 Análise e Cubos OLAP

As ferramentas OLAP [45][46] permitem analisar dados através de uma perspectiva multidimensional, devendo esta análise ser realizada de forma rápida e simples. Com uma ferramenta OLAP, os utilizadores podem fazer *queries* directas aos dados, navegar os dados de forma mais detalhada ou mais resumida (operações de *drill down* e *roll up*), etc. Uma das grandes vantagens deste tipo de ferramentas é a rapidez suposta na obtenção das respostas. Por exemplo, consultas que, numa base de dados relacional, podem demorar horas, passam a ser feitas em segundos.

Outra vertente das análises multidimensionais são os cubos OLAP, estes permitem que os utilizadores naveguem pelos dados de forma simples e objectiva, permitindo que possam agregar informação com mais ou menos detalhe. A operação que permite obter mais detalhe nos dados é a operação de *drill down* e a operação que permite resumir mais os dados, ou seja, obter menor detalhe é a operação de

*roll up*. A Figura 9 mostra uma aplicação (JPivot) que permite fazer análises multidimensionais através de cubos OLAP.

		Measures		
Products	Region	Measures[0]	Measures[1]	Measures[2]
-All Products[0]	-All Region[0]	1,015.22	780.70	890.02
	+All Region[0]			
	+Region[0]	873.77	971.56	1,102.23
	-Region[1]	1,074.22	945.91	823.55
	+Region[1]	876.48	923.64	1,026.13
	+City[0]	1,136.24	825.63	1,067.35
	+City[1]	955.63	1,071.25	1,159.65
	+City[2]	1,163.37	1,015.91	949.34
	+City[3]	909.86	1,136.97	984.30
	+City[4]	1,116.37	1,063.09	985.76
	+City[5]	1,089.23	1,063.38	1,006.19
	+City[6]	1,172.89	992.38	1,050.60
	+City[7]	983.33	834.35	1,131.19
	+Region[2]	1,231.73	1,041.51	1,026.49
	+Region[3]	1,043.67	1,167.71	903.53
	+Region[4]	890.73	1,170.83	955.32
All Products[0]	+Category[0]	1,121.95	957.97	915.73
	-All Region[0]	894.19	1,214.22	1,089.54
	+All Region[0]	1,124.23	1,042.52	925.80
	+Region[0]	904.35	1,106.97	911.88
	+Region[1]	1,086.60	1,031.78	860.41
	+Region[2]	939.83	1,082.63	977.79
	+Region[3]	988.25	850.40	959.15
	+Region[4]			

Figura 9 – Exemplo de ferramenta de OLAP JPivot  
(fonte: <http://JPivot.sourceforge.net/temp-N101F1.html>)

### 2.5.5 Data Mining

*Data Mining* (ou “mineração de dados” em português) [47][48] é um tipo de aplicação de BI muito utilizada e com muita fama entre os utilizadores. Baseia-se num conjunto de processos que têm como objectivo descobrir padrões, tendências e relacionamentos num dado conjunto de dados. Uma aplicação de *data mining* pode ser descritiva e/ou predictiva. *Data mining* descritivo tem como objectivo mostrar os processos e factores relevantes de negócios actuais e específicos. Já o tipo predictivo permite realizar previsões de acordo com os dados e padrões actuais existentes. A Figura 10 mostra um exemplo de aplicação de *data mining*.

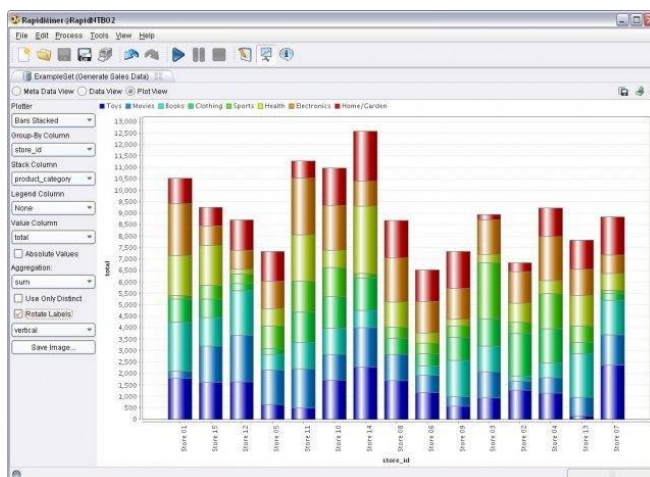


Figura 10 – Exemplo de ferramenta de *data mining* RapidMiner  
(fonte: <http://sourceforge.net/projects/rapidminer/>)

Uma aplicação de *data mining* precisa de intervenção por parte do utilizador final, não é por esta detectar tendências e padrões que, de forma automática, dá ao utilizador o impacto e valor das realidades analisadas, deve haver uma colaboração homem-máquina.

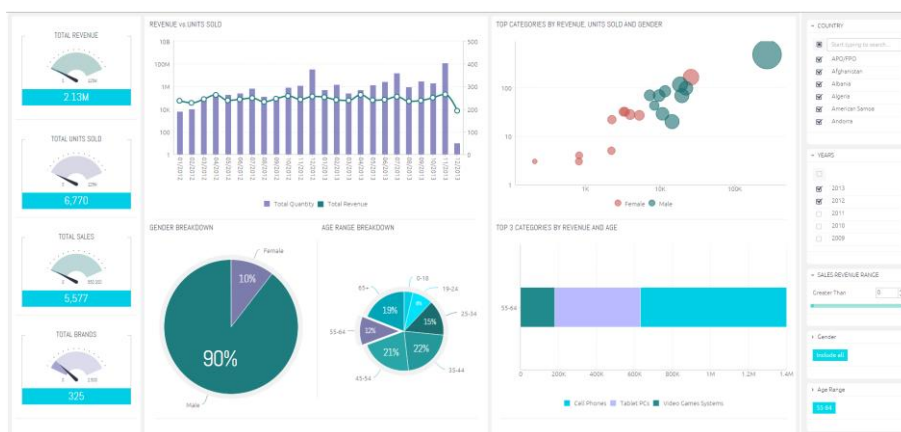
### 2.5.6 Aplicações Avançadas

Considera-se uma aplicação avançada [32] de BI aquela que junta um conjunto de aplicações e ferramentas e disponibiliza ao utilizador uma ferramenta final que têm como grande objectivo auxiliar os decisores de negócio a tomar decisões (o mesmo objectivo da *business intelligence*). Seguidamente serão mostrados alguns casos de estudo.

A aplicação seguinte [49] é um caso de estudo na área de *e-commerce*, ela foi desenvolvida com o intuito de ajudar os decisores de negócio a concluir, por exemplo:

- Evolução do volume de vendas no tempo;
- Melhores compradores/mercados para vários tipos de produtos;
- Quais os tipos de produtos que são comprados por uma pequena quantidade de países/mercados;
- Categorias de produtos mais vendidas.

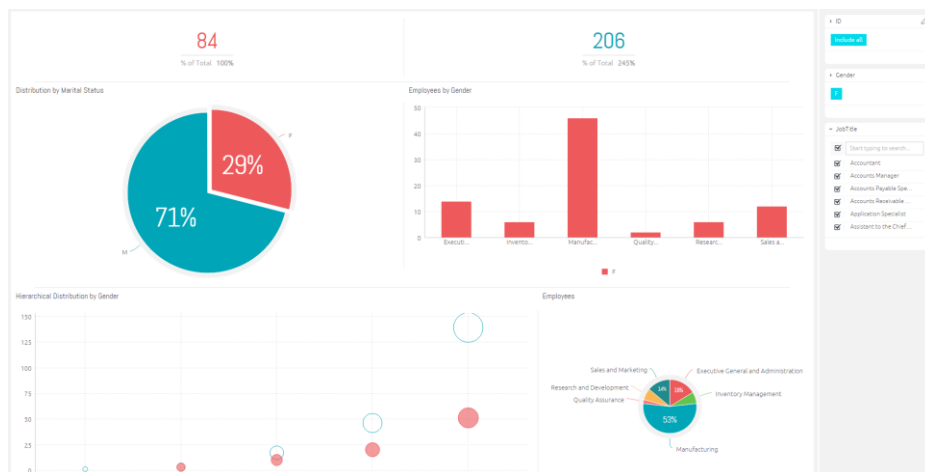
A Figura 11 mostra a plataforma em questão.



**Figura 11 – Plataforma de BI na área de e-commerce**  
(fonte: <http://www-03.ibm.com/software/products/en/business-intelligence>)

A Figura 12 mostra o dashboard conseguido com outra plataforma [49] que foi desenvolvida com o objectivo de mostrar dados qualitativos de colaboradores de uma empresa de telecomunicações. Esta plataforma ajudou os decisores de negócio a concluir, por exemplo:

- Evolução do volume de admissões no tempo;
- Grau de classificação dos colaboradores por área de negócio;
- Quais as áreas de negócio com mais colaboradores.



**Figura 12 – Plataforma de BI na área de recursos humanos**  
(fonte: <http://www-03.ibm.com/software/products/en/business-intelligence>)



## 3. Ferramentas Gratuitas para o Desenvolvimento de Soluções de Business Intelligence

### 3.1 Software Livre

*Software* é um programa de computador que permite que os utilizadores realizem as demais tarefas, desde a escrita de documentos de texto até à visualização de imagens. Sem estes programas, estas tarefas não eram possíveis de realizar informaticamente. Desde à muito que os utilizadores pagam para ter licenças que permitem a utilização dos mesmos sem qualquer problema legal. No entanto, surgiu um novo tipo de *software* que evita o pagamento de licenças para a utilização do mesmo, o *software livre* [50][51]. O *software livre*, de acordo com a *Free Software Foundation* (FSP [52]), é um programa de computador que pode ser executado, copiado, modificado e redistribuído livremente, no entanto podem haver algumas limitações.

As limitações de utilização acima referidas dependem do tipo de licença que o programa tem, quando falamos em *software livre* é importante falar no tipo de *software open source* porque *software* gratuito e *software open source* são coisas diferentes pois não são desenvolvidos tendo em conta as mesmas regras (OSI, 2011). Segundo a *Open Source Initiative* (OSI) um *software open source* respeita algumas das seguintes regras [53]:

- **Distribuição livre:** não deve haver qualquer limitação quanto à venda e distribuição do programa;
- **Acesso ao código fonte:** qualquer pessoa pode aceder gratuitamente ao código fonte ou, em alternativa, o programa deve incluir o código fonte, nem que seja para visualização;
- **Modificações:** não deve existir qualquer limitação à modificação do programa (código fonte ou não) nem à redistribuição do mesmo (desde que seja com a mesma licença original);

- **Sem discriminações:** a utilização do programa não deve ser limitada a qualquer grupo de pessoas (em nome colectivo ou particular) ou tipo de actividade.

As regras anteriores são algumas que devem ser respeitadas para que o *software* possa ser considerado *open source*, quando um programa não pago desrespeita pelo menos uma das regras da OSI então diz-se simplesmente que é gratuito. O *software* gratuito limita muitas vezes os utilizadores quanto à utilização do mesmo, são exemplos as *demos* de programas comerciais (apesar de não serem pagas os utilizadores têm acesso limitado ao número de funcionalidades ou ao tempo de utilização). Porém, existe *software* gratuito que pode ser redistribuído e vendido comercialmente. Não sendo apenas o *software open source* que permite fazer isso. As ferramentas estudadas, e que serão discutidas neste capítulo, não são todas *open source*, nem se pretende apenas programas deste tipo. O objectivo deste trabalho é estudar ferramentas que possam ser utilizadas gratuitamente para uso comercial, sejam ou não *open source*.

As principais vantagens do *software* livre (*software* gratuito e *software open source*) são [54]:

- **Custo reduzido:** o programa é gratuito não existindo custos na obtenção e renovação da licença. Os únicos custos que existem são aqueles que já acrescem os custos das licenças do *software* comercial (equipamentos, luz, etc.);
- **Utilização generalizada:** a utilização dos programas é generalizada, isto é, pode-se utilizar o *software* para vários fins (lucrativos e não lucrativos, lazer, profissionalmente, etc.);
- **Possibilidade de experimentação:** os utilizadores podem experimentar os programas sem qualquer custo, e isto faz com que possa existir comparação e análise crítica entre os programas, ajudando, conseqüentemente, o utilizador a escolher os melhores programas.

As principais desvantagens do *software* livre (*software* gratuito e *software open source*) são [54]:

- **Problemas e bugs:** o facto do programa ser gratuito faz com que as organizações não ganhem dinheiro directamente com este (podem ganhar de outras maneiras, como por exemplo, através de suporte técnico pago, publicidade, etc.) e isto pode fazer com que o empenho e alocação de recursos possa não ser a melhor. O *software open source* é mantido, na maior parte das vezes, por utilizadores globalmente dispersos, são chamadas as comunidades de programadores. Estas não dispõem dos mesmos meios normalmente ao alcance de equipas de *software* contratadas pelas empresas;
- **Falta de documentação e ajuda:** o que foi referido para o ponto anterior também se aplica à documentação e ajuda (instalação, utilização, resolução de problemas, etc.), existem muitos casos em que o suporte não é tão bom e eficaz, comparativamente com as ferramentas comerciais;
- **Durabilidade e longevidade:** nada garante que um programa gratuito e/ou *open source* tenha uma longa duração. Existem muitos projectos deste tipo que acabam por se fundir com outros ou mudar consideravelmente de âmbito num curto espaço de tempo, podendo não ser objecto de melhorias (tornando-se assim obsoleto) ou simplesmente, ser descontinuado.

Existem muitos projectos gratuitos e/ou *open source* que são bem sucedidos e que se apresentam como alternativas viáveis a soluções comerciais pagas. Será que o *software*

*livre* de BI é um destes casos? Este documento visa responder a esta pergunta. Nos capítulos seguintes serão mostradas várias ferramentas livres que permitem criar soluções de BI.

## 3.2 Ferramentas de ETL

Existem várias ferramentas para fazer ETL [33][34][35]. Numa pesquisa efectuada na fase inicial deste trabalho, uma contagem deu como valor algumas dezenas. De entre elas podem referir-se:

- Warehouse Builder (da Oracle);
- Data Migrator (da Information Builders);
- Pentaho Data Integration (da Pentaho);
- DB2 Warehouse (da IBM);
- ETL4ALL (da IKAN);
- Data Integrated ETL (da Sybase);
- Talend Open Studio (da Talend);
- Clover ETL (da OpenSys).

Nem todas as ferramentas anteriores são do tipo *software livre*. Apenas o Talend Open Studio e o Pentaho Data Integration podem ser consideradas como tal. Por isso e devido a informação obtida existentes em diversos fóruns, assim como o número de *downloads* significativo das mesmas, estas duas ferramentas foram escolhidas para serem alvo de estudo.

### 3.2.1 Talend Open Studio

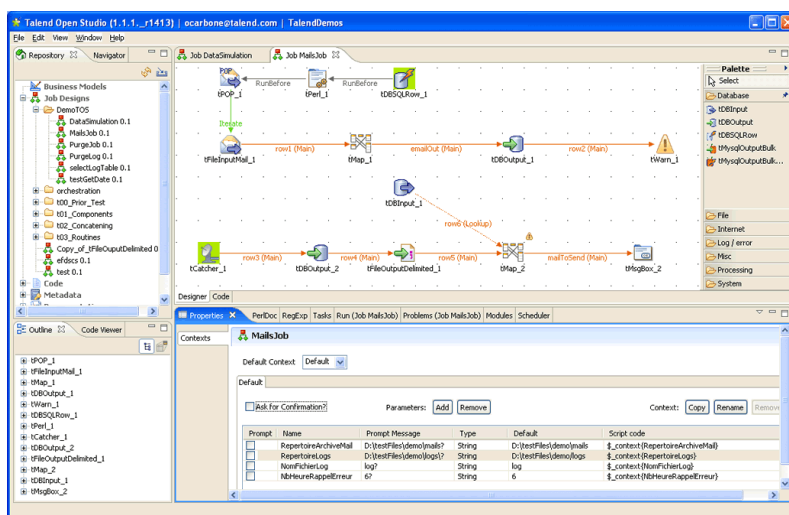
O Talend Open Studio (TOS [55]) é uma ferramenta de ETL desenvolvida pela Talend, uma empresa que nos últimos anos tem crescido muito rapidamente. A Talend diz-se líder no mercado europeu [56] com milhares de *downloads* todas as semanas das suas ferramentas de ETL e *data quality*. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Talend
<b>Download:</b>	<a href="http://www.talend.com/download/talend-open-studio">http://www.talend.com/download/talend-open-studio</a>
<b>Licença:</b>	GNU Lesser General Public License (LGPL) – versão gratuita
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2006
<b>Versão Actual:</b>	6.0.0
<b>Outros Aspectos:</b>	Linguagens de programação usadas são Java e o Perl. Ferramenta compatível com Windows, Unix e Linux.

O TOS é constituído por um conjunto de aplicações para o tratamento de dados, desde o ETL até à *data quality*. Os componentes de ETL e de *data quality* são distribuídos, de forma separada, de duas formas diferentes: uma versão paga e outra gratuita. A versão paga diferencia-se da gratuita no número de funcionalidades disponibilizadas ao utilizador, no entanto estas podem ser consideradas como extras, já que as funcionalidades base são livremente disponibilizadas nas versões gratuitas. Esta ferramenta (componente de ETL) tem um alargado conjunto de utilizadores, pois além de poder ser

utilizada independentemente, esta é incluída noutras ferramentas como por exemplo: o Jaspersoft e o SpagoBI (ferramentas apresentadas neste documento mais à frente).

A Figura 13 mostra o editor do TOS que permite criar os processos de ETL. Como se pode verificar pela ilustração, o editor integra: o editor gráfico (parte central da imagem), editor de configurações (esquerda da imagem), conjunto de componentes responsáveis pelo processo de ETL (parte direita da imagem) e editor de configurações (parte inferior da imagem).



**Figura 13 - Criação de um processo de ETL no Talend Open Studio**  
(fonte: <http://www.informatiquenews.fr/talend-de-letl-aux-outils-hadoop-32206>)

O TOS é uma ferramenta que disponibiliza ao utilizador vários componentes, configurações e metadados para criar processos de ETL. A criação de um processo de ETL é feita através dos *jobs*. Um *job* pode ser visto como uma caixa negra que tem entradas e saídas de dados, e no seu corpo são efectuadas as operações sobre os dados. Um *job* pode agregar outros *jobs* e, com isto um processo de ETL pode ser dividido em diversos *sub-jobs*, cada um tratando de um subprocesso em particular. A criação de jobs é feita através da ligação de vários componentes, uns responsáveis pela extracção de dados, outros pelo seu tratamento e, por fim, os componentes responsáveis pela escrita de dados (seja ficheiro ou base de dados).

O processo de ETL pode ser editado de forma gráfica, mas o Talend, para correr o *job*, usa código Java ou Perl (o utilizador pode escolher qual a sua preferência), e enquanto o utilizador está a desenvolver o *job*, a ferramenta está, em tempo real, a gerar código fonte. Com isto, o utilizador pode ver o código, copiá-lo e corrê-lo num programa isolado. O código fonte é responsável por processar o ETL desenvolvido, e por isso, ao fim do processo desenvolvido, o utilizador para executar o processo de ETL terá que fazer o *deploy* para um servidor que corra código nessa linguagem, e por fim, executar o mesmo.

A organização dos *jobs* no TOS é feita usando *workspaces*. Em cada um podem ser guardados diversos *jobs*, variáveis de contexto, mapeamentos de tabelas, etc. O *workspace* pode ter um projecto ou conjunto de projectos do mesmo âmbito/sistema. Não existe nenhuma limitação quanto à quantidade de *workspaces* que se podem criar. Na Figura 14 pode constatar-se que é necessário escolher sempre um projecto e *workspace* quando se pretendem criar *jobs* ou aceder a *jobs* existentes.

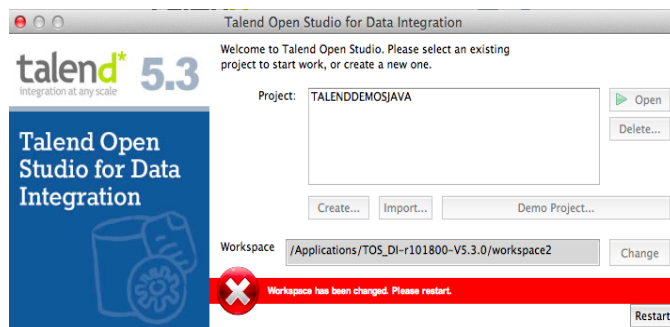


Figura 14 – Seleccção de projecto e *workspace* no Talend Open Studio (fonte: <http://www.talendbyexample.com/talend-project-reference.html>)

O TOS tem mais três características que importa referir, são elas [55]:

- **Metadados:** o potente motor de metadados desta ferramenta permite que sejam guardadas informações sobre tabelas, vistas e procedimentos armazenados de bases de dados. Com esta funcionalidade consegue-se, por exemplo, saber a estrutura física de todas as tabelas e vistas numa base de dados, para posteriormente, quando se pretender utilizar um componente (num *job*) que faça a extracção de dados de uma tabela específica de uma base de dados, basta fazer a conexão e mapeamento iniciais à base de dados e depois arrastar a tabela para o *job*, que automaticamente o *Integrated Development Environment* (IDE) trata de pedir ao utilizador o tipo de componente que deseja utilizar, seleccionando-se depois o componente de extracção e automaticamente o componente trata de extrair dados da tabela fonte desejada;
- **Variáveis de contexto e sessão:** através desta funcionalidade, em cada projecto podem ser definidas variáveis que podem ser reutilizadas e *jobs* diferentes. Por exemplo, pode ser definida uma variável que define qual o ambiente (e.g.: desenvolvimento, *quality review* e produção), e depois, em cada *job*, o utilizador pode definir qual o ambiente no qual o *job* será executado, evitando assim criar um projecto diferente por cada ambiente;
- **Versionamento:** esta funcionalidade permite guardar versões dos *jobs* criados, permitindo ao utilizador criar várias versões do mesmo *job*. Com isto, o utilizador pode por exemplo, recuperar uma versão antiga de um *job*.

#### 3.2.2 Pentaho Data Integration

O Pentaho Data Integration (PDI [57]) nasceu do projecto Kettle [58] que começou por ser um projecto *open source* desenvolvido por uma comunidade de programadores. Mais tarde, a Pentaho ficou responsável pelo projecto e deu-lhe o nome de Pentaho Data Integration. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Pentaho
<b>Download:</b>	<a href="http://community.pentaho.com/projects/data-integration/">http://community.pentaho.com/projects/data-integration/</a>
<b>Licença:</b>	LGPL
<b>Ano de Lançamento:</b>	2000
<b>Versão Actual:</b>	5.4.0
<b>Outros Aspectos:</b>	Linguagem de programação usada é a linguagem Java. Ferramenta compatível com Windows, Unix e Linux.

O PDI é uma ferramenta responsável por processos de ETL. Tal como o Talend, a sua distribuição é feita de duas formas diferentes: uma versão paga e outra gratuita. A versão paga diferencia-se da gratuita, tal como o Talend, no número de funcionalidades disponibilizadas ao utilizador. No entanto estas também podem ser consideradas como extras, já que as funcionalidades base são livremente disponibilizadas na versão gratuita. Esta é a ferramenta de ETL com mais utilizadores no mercado.

O seu editor gráfico é bastante parecido com o TOS, como se pode ver pela Figura 15.

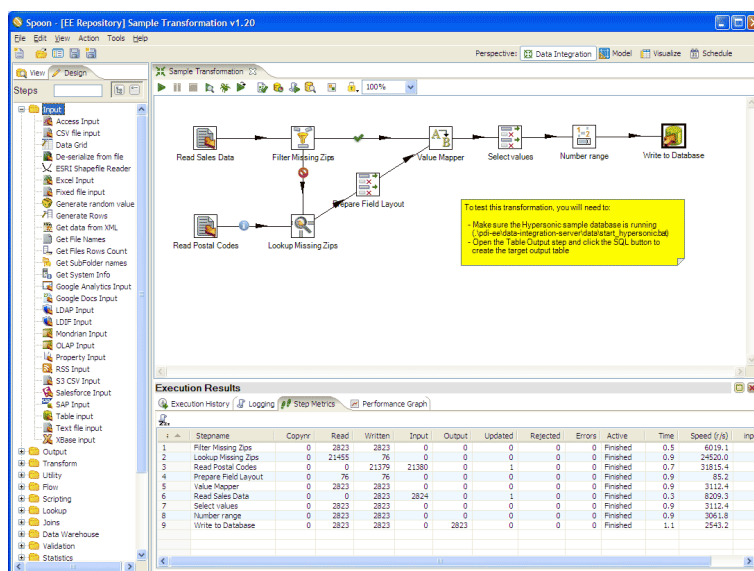


Figura 15 - Criação de um processo de ETL no Pentaho Data Integration (fonte: <http://www.matera.com/br/2012/03/20/integracao-de-dados-na-matera-systems/>)

A criação de funções ETL, tal como no TOS, é feita através de *jobs*, que são constituídos por ligações entre componentes com funções específicas. A forma como são criados os *jobs* é exactamente igual ao TOS. Através de *drag'n'drop* o utilizador pode criar *jobs* complexos, de forma acessível. Existe uma diferença no que concerne ao tipo de código que forma um *job* no PDI comparativamente ao TOS. Enquanto que este último gera códigos em linguagens de alto nível (Java ou Perl), no PDI o código é gerado no formato *eXtensible Markup Language* (XML). Quanto à organização dos desenvolvimentos também se usam *workspaces* para dividir os projectos que contêm os *jobs*. Os conceitos apresentados no capítulo anterior no que toca aos metadados, variáveis de contexto e sessão e versionamento também existem no PDI, a única diferença assinalável é o facto de no TOS poderem ser criadas variáveis de contexto que podem ser reutilizadas noutros projectos do mesmo *workspace*, enquanto que no PDI estas variáveis estão limitadas ao projecto em questão, ou seja, o nível de reutilização é menor do que no TOS.

### 3.2.3 Comparação entre o Talend Open Studio e o Pentaho Data Integration

A Tabela 1 faz uma análise comparativa entre vários aspectos que o autor achou relevantes para permitir uma análise quanto às diferenças entre o TOS e PDI e permitir aquilatar quanto ao seu desempenho quer para a comunidade de programadores, qual a melhor para os utilizadores, etc. Existem muitas outras características que poderiam ser comparados, sendo que se consideraram os seguintes como os mais relevantes [59]:

Tabela 1 - Análise comparativa entre o Talend Open Studio e o Pentaho Data Integration

Características	Talend Open Studio	Pentaho Data Integration
<b>Comportamento</b>	É um gerador de código, no entanto não é requisito que o utilizador conheça a linguagem em questão para criar <i>jobs</i> .	Não gera código de programação, é mais uma ferramenta de transformação do que programação, não está dependente de tipos de dados nativos como o Talend que gera código numa linguagem de programação específica.
<b>Facilidade de utilização</b>	Pouco intuitiva e algo difícil de entender ao início, mas a curva de aprendizagem é muito reduzida. Ao fim de pouco tempo, o utilizador consegue criar os <i>jobs</i> facilmente.	Ferramenta fácil de utilizar e muito intuitiva desde a primeira utilização.
<b>Editor</b>	<i>Interface</i> muito arrumada e que torna fácil encontrar os componentes, metadados, variáveis de sessão e contexto, etc.	<i>Interface</i> confusa que por vezes faz com que a tarefa de encontrar um componente específico se torne penosa.
<b>Ligações a fonte de dados</b>	Ligação a muitas bases de dados, tipos de ficheiros, <i>webservices</i> , etc.	Ligação a muitas bases de dados, tipos de ficheiros, <i>webservices</i> , etc.
<b>Debug dos processos</b>	Bom <i>debug</i> de código, simples e eficaz.	<i>Debug</i> muito básico.
<b>Reutilização de código</b>	Facilita a reutilização de código, pois permite que o utilizador crie os seus próprios componentes.	Muito limitado na reutilização de código.

As duas ferramentas são muito similares, desde a forma como se criam *jobs* até às funcionalidades que oferecem. Através da tabela anterior consegue-se perceber que existe um aspecto em que as duas são iguais, que é a enorme quantidade de ligações que estas ferramentas permitem fazer, existem muitos conectores a vários tipos de Sistemas Gestores de Bases de Dados (SGBD), diversos formatos de ficheiros que podem ser lidos e escritos, ligações a *webservices* de diferentes tipos, etc. A principal diferença no seu modo de funcionamento é que o TOS é um gerador de código e o PDI comporta-se como sendo um transformador que gera ficheiros XML que podem depois ser executados pela própria ferramenta. Assim, pode-se dizer que o PDI não é limitado às características das linguagens de programação como o TOS. Ao nível da utilização, o PDI é mais intuitivo ao início. No entanto, a sua *interface* confusa e pouco arrumada faz com que o TOS ganhe terreno no que toca ao editor, apesar de não ser tão intuitivo ao início, depois do utilizador se sentir confortável com o mesmo, a criação de

*jobs* torna-se mais fácil e rápida. O TOS é mais amigo do programador, principalmente por causa da *debug* e da criação de componentes *customs*, o seu *debug* é potente, permite por exemplo saber, em tempo real, o valor de todas as variáveis num ponto específico do *flow* do *job* e permite que o programador utilize componentes criados pelo próprio, permitindo que o nível de reutilização de código no TOS seja maior do que no PDI.

Em suma, ambas as ferramentas são potentes, muito parecidas e dão resposta ao objectivo a que se propõem que é a extracção, transformação e carregamento de dados. No início, o PDI mostra-se mais fácil de utilizar e a existência de mais tutoriais, comparativamente com o TOS, também ajudou. Mas a utilização continuada mostrou que o TOS apresentou funcionalidades e características que fizeram com que esta ferramenta se tornasse mais querida. O *debug* e a usabilidade do editor fizeram com que o TOS ganhasse a preferência do autor, a estas características juntou-se a possibilidade de se poderem criar componentes próprios, incentivando a reutilização de código e o modo como se criam estes componentes *custom* é fácil e eficaz.

### 3.3 Ferramentas de Reporting

Assim como as ferramentas de ETL [38][39] também existem várias ferramentas para fazer *reporting*, o autor conseguiu encontrar dezenas delas, sendo talvez as mais relevantes:

- Active Reports (da Data Dynamics);
- Crystal Reports (da SAP);
- WebFocus (da Information Builders);
- SQL Server Reporting Services (da Microsoft);
- QlikView (da Qlik);
- BIRT (da Eclipse Foundation);
- JasperReports (da Jaspersoft).

Nem todas as ferramentas anteriores são *software livre*. Apenas o BIRT Project e o JasperReports podem ser consideradas como tal. Por isso e devido a diversos tópicos existentes em vários fóruns, assim como o número de *downloads* significativo das mesmas, estas duas ferramentas foram escolhidas para serem alvo de estudo.

#### 3.3.1 JasperReports

A ferramenta JasperReports [60][61] é uma *framework open source* escrita em Java que permite criar relatórios em diversos formatos (pdf, html, xls, csv, etc.) e que tem muitos recursos que tornam a criação de relatórios fácil para o utilizador final. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Jaspersoft
<b>Download:</b>	<a href="http://sourceforge.net/projects/jasperreports/">http://sourceforge.net/projects/jasperreports/</a>
<b>Licença:</b>	LGPL– versão gratuita
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2002
<b>Versão Actual:</b>	6.1.0

**Outros Aspectos:** Ferramenta compatível com Windows, Unix e Linux.

Os relatórios gerados por esta ferramenta ficam com o formato “.jasper”, mas o conteúdo dos mesmos é construído em XML. Nele estão todas as propriedades do *report* final, e num caso extremo, podem-se criar relatórios apenas com esta linguagem, no entanto, ao fazer isto, os utilizadores não estão a aproveitar o editor existente, o Jaspersoft Studio. A criação de *reports* pode-se fazer de três maneiras: através da edição manual do código XML, através de código java embutido noutras aplicações ou através do editor gráfico referido anteriormente. Já foi referido que existe um editor gráfico que permite a criação dos relatórios, sendo este outro componente do JasperReports. Além da *framework* propriamente dita, existem os seguintes componentes [62]:

- **JasperReports Server:** servidor que permite que sejam criados ou visualizados relatórios em *runtime* em aplicações *desktop* e *web-based*. Este componente não é obrigatório para a utilização da *framework* JasperReports, podendo criarse relatórios com esta última que depois podem ser utilizados directamente pelos utilizadores ou através de *download* feito nas aplicações. Porém, se os utilizadores tiverem a necessidade de pré-visualizar ou interagir directamente com o relatório numa aplicação, então é necessário que este componente seja instalado;
- **Jaspersoft Studio:** editor gráfico que agiliza a criação de *reports*. Trata-se do editor principal a partir da versão 5.5 (actualmente encontra-se na versão 6.1);
- **iReport Designer:** editor gráfico que agiliza a criação de *reports*. Editor obsoleto que ficará sem suporte a partir de 2016.

Durante o teste da *framework* utilizaram-se os dois editores para criar os relatórios. As diferenças entre o Jaspersoft Studio e o iReport Designer são mínimas, sendo ferramentas praticamente iguais. O Jaspersoft Studio destaca-se apenas por ter um aspecto mais limpo e arrumado, mas repita-se, ambas são praticamente iguais e têm exactamente o mesmo fim. O Jaspersoft Studio será o editor que sucederá ao iReport Designer, embora este último ainda esteja disponível para *download*, no futuro deixará de existir. Assim, o Jaspersoft Studio foi o editor utilizado na maioria dos relatórios criados durante a avaliação do editor e da *framework* Jaspersoft. A Figura 16 mostra a criação de um relatório utilizando o editor Jaspersoft Studio.

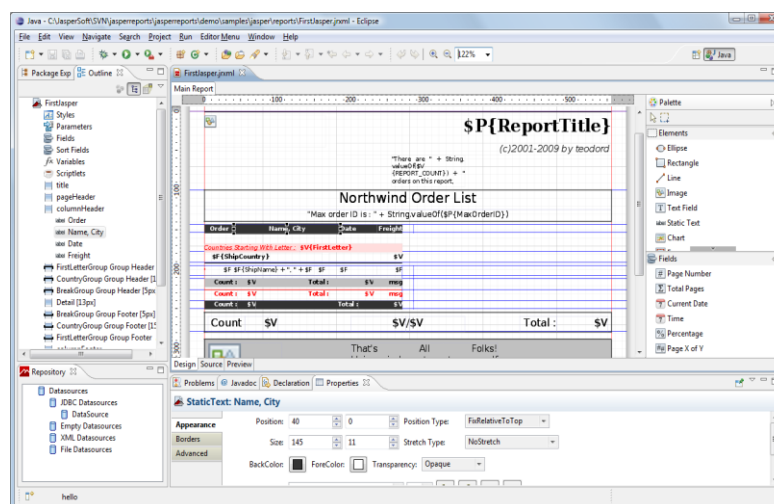


Figura 16 – Criação de um relatório utilizando o Jaspersoft Studio (fonte: <http://marketplace.eclipse.org/content/jaspersoft-studio>)

A criação de relatórios utilizando a *framework* JasperReports e o Jaspersoft Studio é feita através de *drag'n'drop* de componentes de diversos tipos. Segue-se a descrição de cada um destes tipos [61]:

- **Campos:** este tipo de componente tem como principal objectivo fazer o mapeamento de dados. Utilizam-se estes campos para, por exemplo, dizer quais os campos da base de dados que devem ser mostrados numa tabela do *report*;
- **Parâmetros:** este tipo de componente é um *input* do relatório. Se, por exemplo, o relatório dá a possibilidade de mostrar apenas os clientes de um distrito, então o distrito é um parâmetro que o relatório recebe, de maneira a que na consulta à tabela da base de dados exista a possibilidade de filtragem por um campo com valor específico. Os parâmetros são os componentes que permitem que os relatórios não tenham um comportamento estático, passando a ser dinâmicos, pois os utilizadores têm poder para filtrar os dados que aparecem no relatório;
- **Variáveis:** as variáveis são utilizadas para cálculos como: contagem (*count*), soma (*sum*), média (*average*), menor (*lowest*), maior (*highest*), etc. São valores temporários que apenas servem para cumprir determinada tarefa específica no relatório;
- **Expressões:** este tipo de componente é utilizado para formatar o conteúdo de um campo de texto (estático) de forma que este possa possuir valores dinâmicos, são exemplo: data e hora, paginação, textos dos cabeçalhos e rodapés, etc.;
- **Layout:** este tipo de componente lida com as áreas do *report*, que são conhecidas como sendo “secções”, algumas delas: fundo, título, cabeçalho da página, corpo do *report*, rodapé da página, última página do *report*, etc. Cada uma dessas secções tem as suas características.

Através dos componentes anteriores conseguem-se criar relatórios complexos e se forem adicionados parâmetros, os relatórios ficam muito ricos e adicionam valor às análises que os utilizadores poderão vir a fazer com a informação do *report*.

Já o Jaspersoft Server é um módulo interessante. A possibilidade dos utilizadores filtrarem dados e pré-visualizarem os dados que aparecerão no relatório dá uma maior sensação de controlo e interacção com as aplicações que usam estas ferramentas de *reporting*. O JasperReports e os seus demais módulos dão uma solução integrada e interessante para as organizações que querem e/ou precisam de fazer *reporting*.

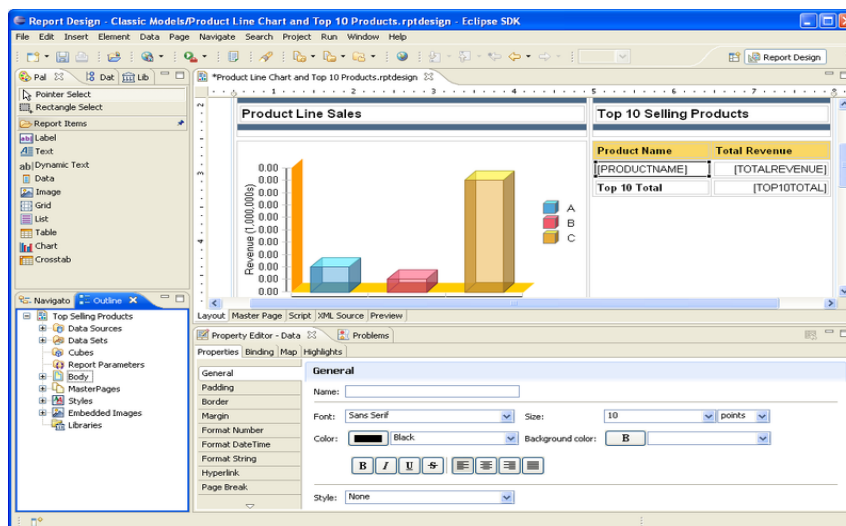
#### 3.3.2 BIRT Project

A ferramenta BIRT Project [63][64] é uma ferramenta *open source* que, tal como o JasperReports, é escrita em Java e permite criar relatórios em diversos formatos (pdf, html, xls, csv, etc.). Tem também muitos recursos que tornam a criação de relatórios uma tarefa acessível para os utilizadores. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	BIRT
<b>Download:</b>	<a href="http://download.eclipse.org/birt/downloads/">http://download.eclipse.org/birt/downloads/</a>
<b>Licença:</b>	Eclipse Public License (EPL) – versão gratuita
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2004
<b>Versão Actual:</b>	4.5.0

**Outros Aspectos:** Ferramenta compatível com Windows, Unix e Linux.

A solução BIRT é uma solução integrada [65]. A criação de relatórios é feita através do BIRT Designer que corre o motor de *reporting*, fazendo assim uma solução de dois em um. A Figura 17 mostra a criação de um relatório utilizando o BIRT Designer.



**Figura 17 – Criação de um relatório utilizando o BIRT Designer**  
(fonte: [https://en.wikipedia.org/wiki/BIRT\\_Project](https://en.wikipedia.org/wiki/BIRT_Project))

A criação de relatórios com o BIRT Designer é feita de uma forma muito semelhante ao JasperReports (utilizando o *designer* Jaspersoft Studio ou o iReport Designer). O utilizador através de *drag'n'drop* arrasta componentes para o corpo do relatório, e os tipos de componentes existentes são iguais ao JasperReports, ou seja os relatórios são constituídos por campos, parâmetros, variáveis, expressões e campos de *layout*.

### 3.3.3 Comparação entre o JasperReports e o BIRT Project

A Tabela 2 faz uma análise comparativa entre vários aspectos que foram seleccionados como os mais relevantes para a análise, de modo a verificar se existem muitas diferenças entre ambas, qual a melhor para os programadores, qual a melhor para os utilizadores, etc. Existem muitas outras características que poderiam ser comparados, o autor achou os seguintes os mais relevantes [66].

**Tabela 2 - Análise comparativa entre o JasperReports e o BIRT Project**

Características	JasperReports	BIRT Project
<b>Impressão de relatórios</b>	Relatórios otimizados para impressão ( <i>pixel perfect</i> ) mas também podem ser utilizados em aplicações <i>desktop</i> e <i>web-based</i> .	Construído para aplicações <i>web</i> , não tendo tanto foco no formato em papel.
<b>Editores</b>	Ambos os editores gráficos (Jaspersoft Studio e iReport Designer) são lentos,	Editor mais rápido e com menos <i>bugs</i> do que os

	<p>havendo muitos <i>crashes</i> durante a edição (especialmente quando se trabalha com imagens);</p> <p>Editores não são intuitivos, porém, dispõem muitas funcionalidades que permitem um controlo total sobre os relatórios a criar.</p>	<p>editores do JasperReports;</p> <p>Editor muito intuitivo e fácil de utilizar, não tem tantas funcionalidades para ajudar na criação dos relatórios, mas as funcionalidades de que dispõe são suficientes para criar relatórios complexos.</p>
<b>Sub-Reports</b>	<p>Suporta o conceito de <i>sub-report</i>, isto é, podem-se criar mini-relatórios que podem ser usados nos <i>master reports</i>. O uso da máxima <i>divide to conquer</i> promove a reutilização.</p>	<p>Não suporta o conceito de <i>sub-report</i>. Existe apenas um documento onde são colocados todos os componentes do relatório.</p>
<b>Layout e estilos</b>	<p>Permite a personalização total do <i>layout</i> do relatório através de CSS (Cascading Style Sheets).</p>	<p>Permite a personalização do <i>layout</i> utilizando o CSS, mas de uma forma mais limitada (não permite, por exemplo, a herança de estilos).</p>
<b>Múltiplas fontes de dados</b>	<p>A utilização de mais do que uma fonte de dados é um pesadelo. Os <i>sub-reports</i> têm que usar a fonte de dados do <i>master report</i>, e se este último não tiver nenhuma é preciso recorrer a certos “truques” para ultrapassar as dificuldades.</p>	<p>A utilização de várias fontes de dados é fácil e sublime</p>
<b>Exportação</b>	<p>Permite a exportação para diversos formatos (pdf, excel, word, imagem, etc.).</p>	<p>Permite a exportação para diversos formatos (pdf, excel, word, imagem, etc.).</p>

Em resumo, a tabela anterior permite verificar que têm em comum os diversos formatos disponíveis para a exportação dos relatórios, sendo que a criação gráfica dos relatórios é também muito parecida. O JasperReports permite ainda a criação e edição de relatórios usando apenas XML, no entanto, o utilizador pode demorar bastante mais tempo a criar um relatório se optar por esta última opção, e na maioria dos casos, os utilizadores acabam sempre por utilizar os editores gráficos para a criação e edição de qualquer tipo de documento. Os editores do JasperReports são um problema ao nível da usabilidade e facilidade de uso, os vários *bugs* que os editores têm são mais uma agravante assinalável. O editor BIRT Designer é simples, fácil e eficaz, destacando-se claramente pela positiva face aos editores do JasperReports. A facilidade com que se usam várias fontes de dados no BIRT Designer é um factor importante que faz com que este se destaque positivamente face aos editores do JasperReports. Os relatórios do JasperReports são mais bonitos e apresentam uma qualidade final

superior, muito devido à orientação para a impressão que a ferramenta tem. O suporte dos *sub-reports* facilitam muito a vida aos utilizadores do JasperReports, no BIRT Designer, por vezes, torna-se difícil criar relatórios complexos pois apenas existe um *master report*.

Em suma, as ferramentas são muito idênticas na forma de criar os relatórios, mas entre as duas o autor opta pelo BIRT Project. O BIRT Project tem apenas um editor que já corre o motor de *reporting*, logo não é necessário instalar vários componentes como no JasperReports, mas o principal motivo de preferência é a simplicidade e usabilidade que o editor tem, já para não falar dos *bugs* dos editores do JasperReports que por vezes faz com que se tenha que começar a fazer algo de novo pois o editor simplesmente deixa de funcionar.

## 3.4 Ferramentas de OLAP

Existem também muitas ferramentas na área do OLAP [45][46]. Seguem-se algumas:

- Mondrian (desenvolvido pela Universidade de Utrecht);
- SQL Server Analysis Services (da Microsoft);
- Palo OLAP Server (da Jedox);
- Business Objects (da SAP);
- Cognos (da IBM);
- JPivot (desenvolvida por Andreas Voss).

Das ferramentas anteriores as únicas que são *software livre* é o Mondrian e o JPivot por isso e devido a diversos tópicos existentes em diversos fóruns, assim como o número de *downloads* significativo das mesmas, estas duas ferramentas foram escolhidas para serem alvo de estudo.

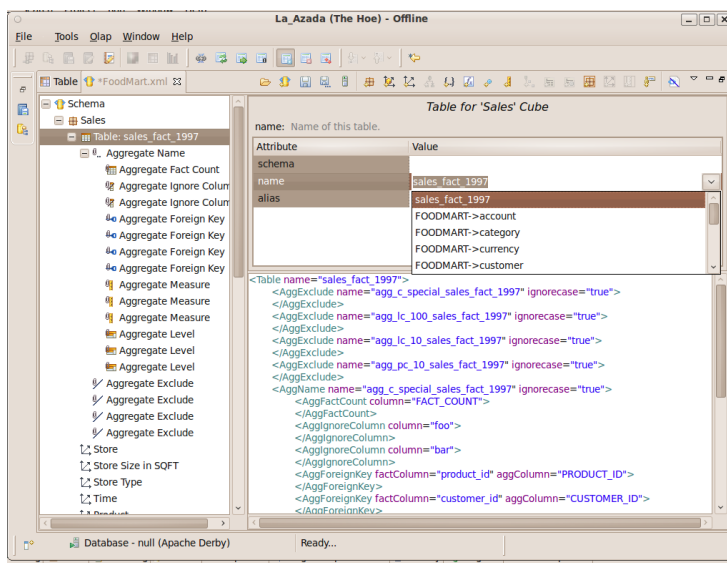
### 3.4.1 Mondrian

A ferramenta Mondrian [67] é um servidor OLAP que permite a criação de cubos de dados e, conseqüentemente, o cruzamento de dados de vários factos e dimensões. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Pentaho
<b>Download:</b>	<a href="http://sourceforge.net/projects/mondrian/files/mondrian/mondrian-3.10.0/mondrian-3.10.0.1-130.jar/download">http://sourceforge.net/projects/mondrian/files/mondrian/mondrian-3.10.0/mondrian-3.10.0.1-130.jar/download</a>
<b>Licença:</b>	EPL
<b>Tem Versão Paga:</b>	Não
<b>Ano de Lançamento:</b>	2002
<b>Versão Actual:</b>	3.10.0
<b>Outros Aspectos:</b>	Ferramenta compatível com Windows, Unix e Linux.

Esta ferramenta foi desenvolvida pela Pentaho usando linguagem Java, recorre a outras linguagens para fazer as análises, como por exemplo: a linguagem de modelação *MultiDimensional eXpression* (MDX [68]), XML for Analysis [69] e *Java Online Analytical Processing* (JOLAP [70]). O Mondrian utiliza o conector *Java Database Connectivity* (JDBC [71]), logo a quantidade de SGBD's suportados

é enorme, podendo-se escolher praticamente qualquer base de dados existente para alimentar o servidor Mondrian. A Figura 18 mostra a edição de um cubo de dados no Mondrian Schema Editor.



**Figura 18 – Editar cubos de dados na Mondrian Schema Editor**  
(fonte: [http://azada.sourceforge.net/schema\\_workbench.html](http://azada.sourceforge.net/schema_workbench.html))

O Mondrian é apenas um servidor OLAP e embora exista o editor Mondrian Schema Editor, este apenas permite criar os cubos de dados e fazer consultas utilizando a linguagem MDX, para visualizar a informação e fazer operações OLAP (pe: *drill down*, *roll-up*, *slice and dice*, etc.) é necessário recorrer a outras ferramentas, como por exemplo o JPivot que será analisado no subcapítulo seguinte.

Como já foi referido, a linguagem MDX é utilizada para consultar e criar cubos de dados, veja-se a diferença entre esta linguagem e o SQL [68]:

### SQL

```
SELECT dpc.EnglishProductCategoryName ,
       SUM(SalesAmount) AS SalesAmount
FROM   dbo.FactInternetSales fis
       INNER JOIN dbo.DimProduct dp ON dp.ProductKey = fis.ProductKey
       INNER JOIN dbo.DimProductSubcategory dps ON dps.ProductSubcategoryKey =
       dp.ProductSubcategoryKey
       INNER JOIN dbo.DimProductCategory dpc ON dpc.ProductCategoryKey =
       dps.ProductCategoryKey
GROUP BY dpc.EnglishProductCategoryName
```

### MDX

```
SELECT
[Measures].[Internet Sales Amount] ON COLUMNS,
[Product].[Product Categories].[Category] ON ROWS
FROM [Adventure Works]
```

Ambas as consultas anteriores têm o mesmo objectivo que é retornar o total de vendas agrupadas por categoria de produtos. Como se pode constatar, as cláusulas principais (select, from, etc.) continuam iguais. No entanto, o código necessário para consultar os mesmos dados é substancialmente diferente,

mais simples utilizando MDX. A linguagem MDX agrupa automaticamente os dados e diferencia as colunas das linhas. Já em SQL, é necessário fazer várias junções entre a tabela de factos e as dimensões para produzir o mesmo resultado. Em MDX, estando o servidor OLAP com o cubo de dados correctamente configurado, as relações entre os factos e as dimensões são automaticamente detectadas.

#### 3.4.2 JPivot

Na apresentação do Mondrian foi dito que esta ferramenta era apenas um servidor OLAP, e conseqüentemente, não tinha nenhuma funcionalidade de visualização de dados em tabela ou gráfico com opções de filtragem, operações de OLAP, etc. Existem diversas ferramentas que têm como função apresentar e permitir que o utilizador tenha interacção com os dados que vêm do servidor OLAP (Mondrian ou outro). Uma dessas ferramentas, e talvez a mais conhecida e com mais tempo de mercado, é o JPivot [72]. O JPivot é uma *Application Programming Interface* (API [73]) feita em *Java Server Pages* (JSP [74]). Esta ferramenta permite a criação de tabelas e gráficos de dados para fazer OLAP, dispondo de várias funcionalidades como por exemplo a filtragem de dados e a execução de operações OLAP (*drill-down*, *roll-up*, *slice and dice*, etc.). Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Pentaho
<b>Download:</b>	<a href="http://sourceforge.net/projects/JPivot/">http://sourceforge.net/projects/JPivot/</a>
<b>Licença:</b>	Commercial Pilot Licence (CPL) 1.0
<b>Tem Versão Paga:</b>	Não
<b>Ano de Lançamento:</b>	2003
<b>Versão Actual:</b>	1.8.0
<b>Outros Aspectos:</b>	Ferramenta compatível com Windows, Unix e Linux.

O JPivot utiliza o servidor Mondrian e a linguagem XML. A como ferramentas de suporte às consultas e análises OLAP. O JPivot dá aos utilizadores a capacidade de [73]:

- **Visualizar dados em tabelas:** os dados visualizados no JPivot aparecem sempre primeiro em formato de tabela, é nestas tabelas que se podem depois fazer outras operações de navegação;
- **Geração de gráficos:** é possível gerar um gráfico com os dados da consulta actual, mesmo depois de feitas operações de filtragem, *drill-down*, *roll-up*, etc.;
- **Fazer consultas directas à base de dados:** é permitido que os utilizadores consigam fazer consultas directamente ao servidor OLAP através da linguagem MDX;
- **Fazer operações OLAP:** é possível fazer as operações típicas de OLAP, como por exemplo: *drill-down*, *roll-up*, *slice and dice*, etc.;
- **Exportar as consultas:** o utilizador pode exportar os dados para o formato pdf e excel.

A Figura 19 mostra a ferramenta JPivot, onde se pode observar que os dados são mostrados a partir de tabelas. Nestas conseguem-se fazer operações de *drill-down* e *roll-up* (através dos botões de “+” e “-“, para respectivamente, mostrar mais detalhe ou menos detalhe), o utilizador pode filtrar dados das colunas e das linhas e fazer outras operações recorrendo ao menu da ferramenta.

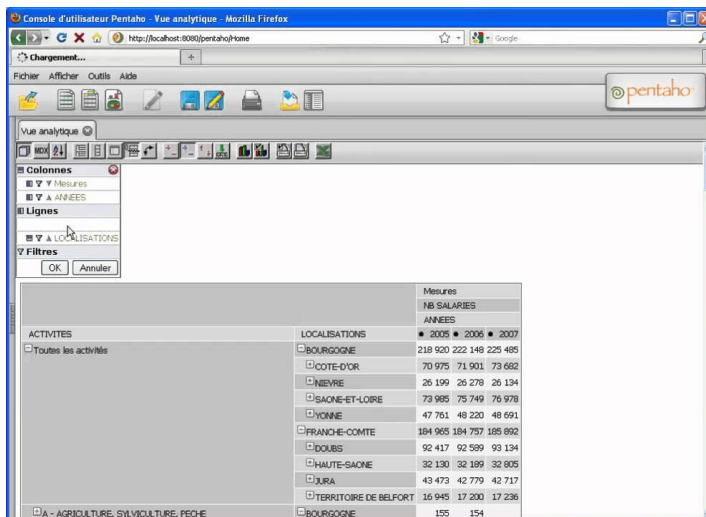


Figura 19 – Exemplo de utilização da ferramenta JPIVOT (fonte: <http://pentaho.dlpage.phi-integration.com/mondrian>)

O JPIVOT tem um menu que permite ao utilizador fazer diversas acções sobre os dados. A Figura 20 seguinte mostra as funcionalidades do menu:

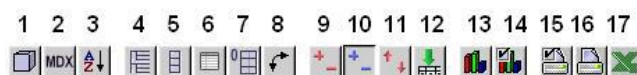


Figura 20 – Menu do JPIVOT

Cada botão do menu faz o seguinte:

1. Seleccionar a consulta pretendida, especificando as dimensões e medidas que serão alvo de análise;
2. Consultar directamente a base de dados utilizando a linguagem MDX;
3. Ordenar os dados;
4. Visualizar e ocultar os membros-pai dentro da hierarquia dos dados;
5. Ocultar registos vazios;
6. Mostrar as propriedades das dimensões do cubo (se for aplicável aos dados em análise);
7. Ocultar colunas ou linhas vazias;
8. Fazer operação de *rotate*, ou seja, trocar as colunas pelas linhas;
9. Fazer *drill down* ou *roll up* completo do elemento/dado seleccionado, ou seja, expande todos os nós onde exista o dado específico;
10. Fazer *drill down* ou *roll up* incompleto do elemento/dado seleccionado, ou seja, expande apenas os níveis onde exista o dado específico;
11. Fazer *drill replace* de o elemento/dado seleccionado, ou seja, substitui a raiz da dimensão actual pelo dado específico;
12. Fazer *drill through* de o elemento/dado seleccionado, ou seja, mostra os dados de detalhe relativo ao dado agregado seleccionado;
13. Mostrar ou ocultar gráfico dos dados que estão a ser usados na consulta actual;
14. Configurar gráfico a gerar, conseguindo-se configurar o tipo (linhas, barra vertical, barra horizontal, 3D), altura e largura, entre outras;
15. Configurar propriedades de impressão;
16. Exportação das tabelas e gráficos para o formato pdf;

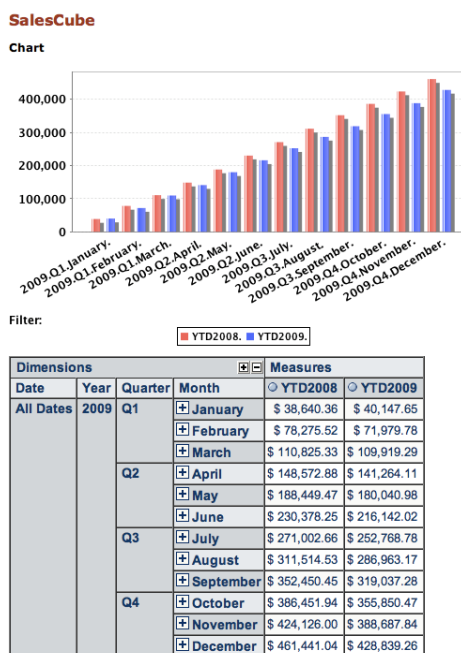
17. Exportação das tabelas para o formato de excel;

Foi referido anteriormente que era possível fazer consultas utilizando consultas MDX, veja-se a *query* seguinte [68]:

```
WITH
MEMBER Measures.YTD2009 as 'Aggregate(YTD(Date.CurrentMember), Measures.Turnover) '
MEMBER Measures.YTD2008 as 'Aggregate(YTD(ParallelPeriod(Date.Year, 1,
Date.CurrentMember)), Measures.Turnover) '

SELECT {
    [Measures].YTD2008,
    [Measures].YTD2009
} ON COLUMNS,
Descendants(Date.[2009], Date.Month) ON ROWS
FROM [Sales]
```

Através da consulta anterior o JPivot mostra a tabela e o gráfico da Figura 21.



**Figura 21 – Exemplo de consulta feita no JPivot**  
 (fonte: <http://type-exit.org/adventures-with-open-source-bi/2010/07/a-simple-date-dimension-for-mondrian-cubes/>)

Como se pode constatar pela figura anterior, o JPivot correu a *query* MDX directamente no servidor OLAP (neste caso o Mondrian) e colocou os dados sobre o formato de tabela. Complementando esta, aparece ainda um gráfico de barras.

### 3.4.3 Comparação entre o Mondrian e o JPivot

Não se deve fazer uma comparação entre o Mondrian e o JPivot, apesar de ambas serem ferramentas OLAP, têm propósitos diferentes. O Mondrian é um servidor OLAP e o JPivot é uma API que mostra os dados de forma gráfica utilizando o Mondrian como servidor OLAP. A Figura 22 mostra a relação entre estas duas ferramentas.

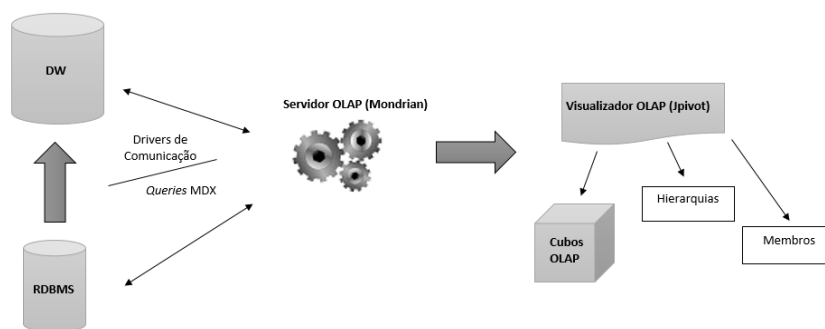


Figura 22 – Colaboração entre o Mondrian e o JPivot

Como se pode ver na figura anterior, o servidor OLAP (Mondrian) consulta directamente as fontes de dados (neste caso as fontes de dados representadas são bases de dados relacionais ou *data warehouses*, mas podiam ser de outro tipo), depois cria os modelos de dados, incluindo os cubos OLAP. Por fim, o JPivot trata de mostrar os dados aos utilizadores de uma forma gráfica. Não se pode então comparar estas duas ferramentas pois elas complementam-se uma à outra.

### 3.5 Ferramentas de Data Mining

Existem também muitas ferramentas na área de *data mining* [47][48]. O autor conseguiu encontrar as seguintes:

- SPSS Modeler (da IBM);
- SQL Server Analysis Services (da Microsoft);
- Data Mining (da Oracle);
- Weka (desenvolvida pela Universidade de Waikato);
- R Programming Language (desenvolvida pelo GNU Project);
- RapidMiner (da RapidMiner).

Das ferramentas anteriores as únicas que podem ser usadas como *software livre* é a Weka, a R Programming Language e o RapidMiner. O autor não teve oportunidade de experimentar todas, a ferramenta Weka e o Rapid Miner foram as escolhidas por serem bastante discutidas em fóruns de utilizadores.

#### 3.5.1 Weka

A ferramenta Weka [75] é uma ferramenta que agrega vários algoritmos de inteligência artificial que têm como objectivo extrair conhecimento dos dados analisados. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Desenvolvida inicialmente pela Universidade de Waikato (Nova Zelândia) e adquirida em 2006 pela Pentaho
<b>Download:</b>	<a href="http://sourceforge.net/projects/weka/">http://sourceforge.net/projects/weka/</a>
<b>Licença:</b>	General Public License (GPL)
<b>Tem Versão Paga:</b>	Não
<b>Ano de Lançamento:</b>	1993 pela Universidade de Waikato, posteriormente em 2006 pela Pentaho

**Versão Actual:** 3.6.12  
**Outros Aspectos:** Ferramenta compatível com Windows, Unix e Linux.

A Weka foi desenvolvida em linguagem Java, implementa algoritmos de inteligência artificial e disponibiliza algumas funcionalidades como [76]:

- **Previsão:** a Weka disponibiliza algoritmos que permitem o computador aprender, isto é, através de padrões existentes nos dados à data da análise, o computador consegue prever possíveis acontecimentos futuros
- **Classificação e regressão:** a ferramenta implementa algoritmos de classificação e regressão, estes, tal com os algoritmos de previsão, têm como objectivo prever futuras ocorrências de dados através dos padrões actuais
- **Clustering:** ao aplicar algoritmos de *clustering* os utilizadores conseguem identificar dados que partilham tendências e padrões semelhantes

A Figura 23 mostra como se pode aplicar *clustering* a um conjunto de dados, na figura conseguem-se ver um conjunto de dados em gráficos, e os padrões dos dados estão evidenciados através de cores diferentes, através da dispersão dos dados consegue-se extrair conhecimento sobre cada grupo de dados (*cluster*).

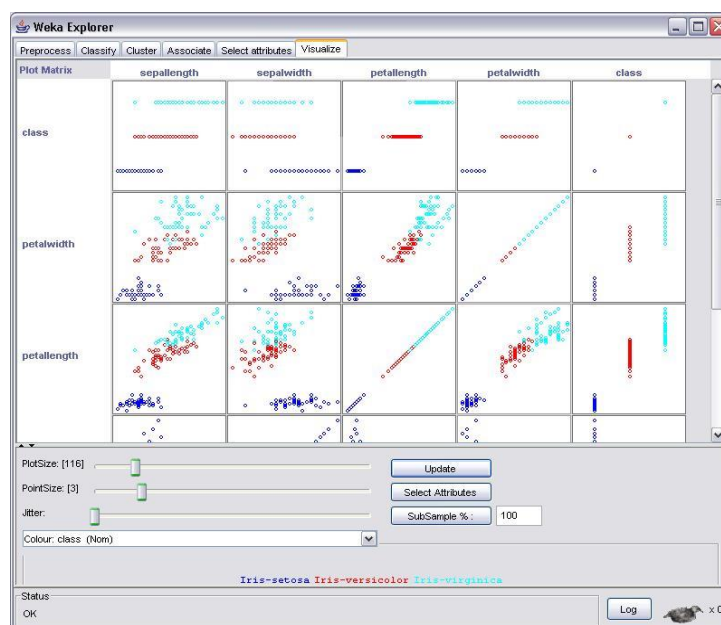


Figura 23 – Exemplo de aplicação de *clustering* utilizando a ferramenta Weka (fonte: <http://research.cs.queensu.ca/home/cisc333/tutorial/Weka.html>)

#### 3.5.2 Rapid Miner

A ferramenta Rapid Miner [77] é uma ferramenta que disponibiliza várias funcionalidades de *data mining* e outras (ETL, *reporting*, etc.). Segue-se o cartão-de-visita desta ferramenta:

**Responsabilidade:** Desenvolvida inicialmente pela Universidade de Dortmund (Alemanha) e adquirida posteriormente pela Rapid-I

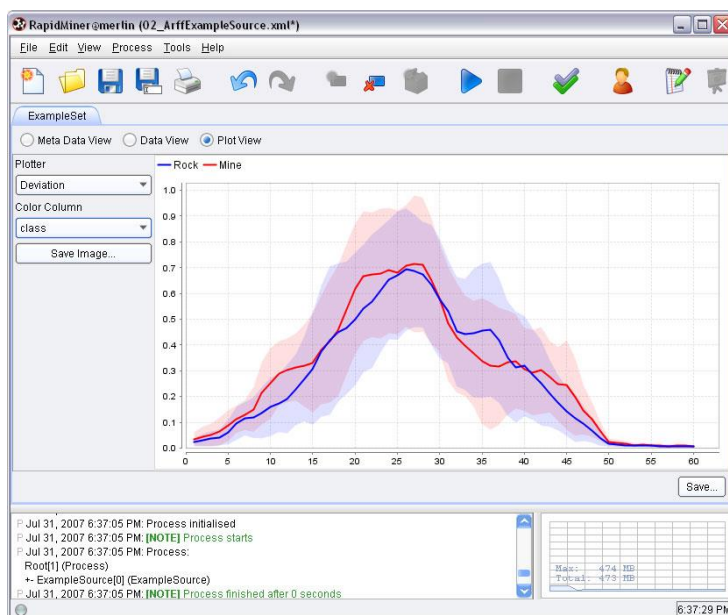
**Download:** <http://sourceforge.net/projects/rapidminer/>

<b>Licença:</b>	Affero General Public License (AGPL)
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2006
<b>Versão Actual:</b>	5.3.013
<b>Outros Aspectos:</b>	Ferramenta compatível com Windows, Unix e Linux.

Esta ferramenta é a continuação do projecto *Yet Another Learning Environment* (YALE [78]) desenvolvido pela Universidade de Dortmund. Após ser adquirida pela Rapid-I, passou a ter versões pagas. No entanto, dado o objectivo do presente trabalho, a versão analisada foi a última versão *open source* disponibilizada pela comunidade. Algumas funcionalidades disponibilizadas são [77]:

- Análises OLAP;
- ETL;
- *Reporting*;
- *Data Mining*.

Apesar de fazer *reporting*, análises OLAP, ETL, entre outras, o foco desta ferramenta é o *data mining*, e todas as funcionalidades que vão além do *data mining* têm como objectivo complementá-lo esta mesma funcionalidade, considerou-se que o Rapid Miner é uma ferramenta de *data mining* e não um pacote integrado de soluções, apesar desta última classificação também se aceitar. Esta ferramenta foi desenvolvida em linguagem Java, e no seu *core* de *data mining* utiliza o motor do Weka mais a linguagem de programação R. Os sistemas fonte de dados disponíveis no Rapid Miner vão desde os ficheiros excel, passando por ficheiros de texto, até aos SGBD mais conhecidos (e.g.: Microsoft SQL Server, Oracle, MySQL, Sybase, etc.). A Figura 24 mostra como se pode user o *data mining* do RapidMiner, aplicando: *reporting* e *dashboarding*. Como já foi referido, o foco desta ferramenta são os processos de *data mining*. A figura analisa padrões actuais de dados, utilizando para representar esses padrões um gráfico que pode depois ser inserido num *report* ou *dashboard*.



**Figura 24 – Exemplo de aplicação de *data mining* utilizando o RapidMiner (fonte: <http://www.pressebox.com/attachments/details/62678>)**

### 3.5.3 Comparação entre o Weka e o Rapid Miner

Não se deve fazer uma comparação directa entre o Weka e o RapidMiner no que toca às funcionalidades de *data mining*. O Rapid Miner disponibiliza várias funcionalidades para lá do *data mining*, e nesta última utiliza os algoritmos do Weka, ou seja, o Weka é utilizado no Rapid Miner. No entanto existem características gerais que podem ser alvo de comparação entre ambas. A Tabela 3 mostra as principais [79].

Tabela 3 - Análise comparativa entre o Weka e o Rapid Miner

Características	Weka	Rapid Miner
<b>Criação de flows</b>	Não oferece <i>suggests</i> para corrigir erros do utilizador na criação de processos.	Editor sugere alterações quando uma parte de um <i>flow</i> de análise está incorrecto.
<b>Aprendizagem</b>	O facto de existirem poucos <i>tutoriais</i> , torna difícil, aos iniciantes criar processos de análise.	Muita documentação e ajuda <i>online</i> .
<b>Análises</b>	Fácil comparar diferentes algoritmos para o mesmo problema/análise.	Difícil a comparação de algoritmos diferentes.
<b>Leitura de ficheiros</b>	Lentidão em ler ficheiros excel.	Melhor ligação a ficheiros excel e o <i>reader</i> de ficheiros csv é mais robusto e rápido.

Os utilizadores que apenas pretendam utilizar funcionalidades de *data mining* de dados podem encontrar no Weka uma solução completa que oferece muitos algoritmos. Um dos aspectos mais importantes a destacar é a forma fácil como se podem comparar resultados de uma análise, utilizando diferentes algoritmos simultaneamente. O Rapid Miner não é só uma ferramenta de *data mining*, apesar do foco da ferramenta ser esta área, como já foi referido, ele oferece mais funcionalidades como *reporting*, *dashboarding*, ETL, entre outras, tendo ainda um editor muito simpático e usável, comparativamente com o Weka. No entanto, como tem várias funcionalidades, o foco no *data mining*, por parte do utilizador, pode-se perder e a comparação dos resultados de algoritmos diferentes pode ser uma tarefa morosa. De todas as ferramentas estudadas, estas duas foram as que tiveram menos tempo de avaliação, por este motivo, não existe nenhuma preferência por uma delas.

### 3.6 Ferramentas Suite

Existem muitas ferramentas que permitem fazer BI. As descritas atrás dedicam-se a uma área em particular, como por exemplo: ETL, *dashboarding*, *reporting*, OLAP, etc. As ferramentas do tipo *suite*

são soluções integradas que disponibilizam várias funcionalidades, de áreas diferentes, no mesmo *package*. Como será analisado nas secções seguintes, algumas destas *suites* utilizam ferramentas independentes na sua solução geral, isto faz com que estas ferramentas sejam mais funcionais. No entanto, são também mais dependentes das soluções particulares. Na maioria dos casos ter uma ferramenta *suite* é mais vantajoso do que ter diversas ferramentas particulares e específicas de uma área. Com uma *suite* evitam-se os arquipélagos de funcionalidades, em vez de cada ilha ter uma funcionalidade específica, juntam-se todas e forma-se uma única ilha de funcionalidades, sendo isto vantajoso para os utilizadores pois ficam com uma solução *all-in-one*. No mercado actual existem muitas *suites* que permitem fazer BI. De acordo com a pesquisa do autor as mais relevantes para analisar, e que respeitam o objectivo de fazer BI *low-cost*, são: o TIBCO Jaspersoft, o Actuate BIRT, o Jedox PALO, o Pentaho e o Spago BI. Estas serão o alvo de uma análise cuidada e detalhada.

#### 3.6.1 TIBCO Jaspersoft

A ferramenta Jaspersoft [80] nasceu do projecto JasperReports. A empresa TIBCO adquiriu os direitos deste último e passou a incluir o seu motor nas suas soluções. A TIBCO, no seu próprio *website*, diz-se líder no mercado de *business intelligence*, com milhões de *downloads* conseguidos com o Jaspersoft e que em 2013 ganhou o prémio de melhor solução para BI da InfoWorld (versão comercial) [81]. A solução Jaspersoft tem uma versão gratuita e outras quatro comerciais/pagas. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Jaspersoft
<b>Download:</b>	<a href="http://community.jaspersoft.com/project/jasperreports-library">http://community.jaspersoft.com/project/jasperreports-library</a>
<b>Licença:</b>	LGPL – versão gratuita
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2002
<b>Versão Actual:</b>	6.1.0
<b>Outros Aspectos:</b>	Ferramenta compatível com Windows, Unix e Linux.

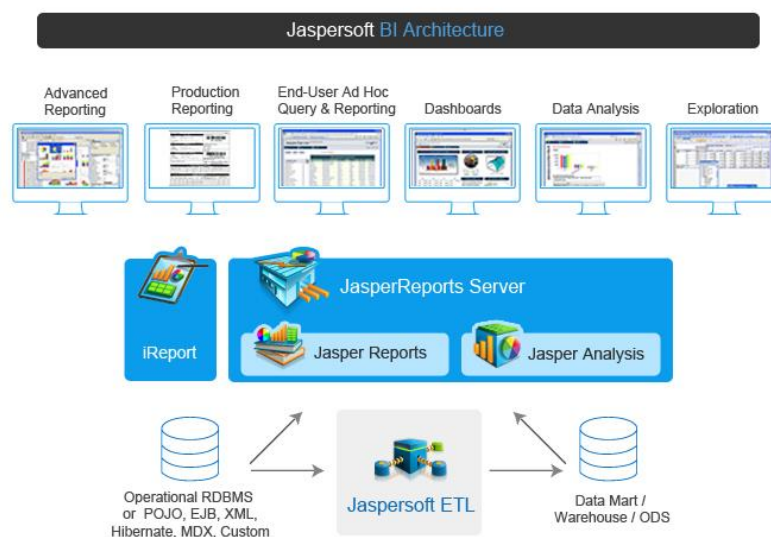
O Jaspersoft, versão gratuita/*community*, continua a ser o projecto JasperReports, um projecto *open source* mantido por uma comunidade de utilizadores. Tudo o que foi dito na secção 3.3.1 aplica-se para o Jaspersoft versão gratuita/*community*. A única diferença que existe antes e depois da compra dos direitos por parte da TIBCO, é que foi adicionada a funcionalidade de ETL. A empresa, em colaboração com a Talend, desenvolveu *drivers* que permitem que o Talend Open Studio consiga fazer *upload* directo para o Jaspersoft Server, ou seja, a versão gratuita/*community* também tem a funcionalidade de ETL, mas esta funcionalidade é simplesmente a utilização do Talend Open Studio.

Apesar do JasperReport já ter sido analisado anteriormente, a versão profissional do Jaspersoft foi analisada durante o período experimental de trinta dias apesar de fugir dos objectivos e do âmbito deste documento, existem características muito interessantes nesta versão comercial (que poderá interessar às organizaçõess que utilizem ferramentas livres mas que estão dispostas a fazer o “*upgrade*” para versões comerciais), muito resumidamente, destacam-se as seguintes [82]:

- **Ad-Hoc Reports:** possibilidade dos utilizadores, sem conhecimentos técnicos, criarem os seus próprios relatórios através de *drag’n’drop*;

- **API Visualize.js:** API em javascript que torna possível, de modo muito simples e rápido, a integração de documentos do Jaspersoft (*reports, dashboards, etc.*) em aplicações *web* personalizáveis e escaláveis;
- **Dashboards:** vários tipos de gráficos e tabelas permitem criar *dashboards* muito elegantes e usáveis;
- **Análise em memória:** análises de dados em *runtime* com *performances* de velocidade impressionantes. Os utilizadores podem ver dados de bases de dados relacionais e multidimensionais em “directo”, é possível ainda fazer *filtragens, pivoting, charting* e outras, enquanto se analisam os dados em tempo real;
- **Análises interactivas:** disponibilidade de os utilizadores interagirem com os dados. Os gráficos são o maior exemplo disso, sendo possível, por exemplo, que o utilizador carregue numa parte de um gráfico e as tabelas que mostram os dados refrescaram automaticamente, de acordo com a categoria escolhida.

A Figura 25 mostra a distribuição dos diversos módulos que compõem a arquitectura do Jaspersoft. O iReport, o JasperReports Server, o JasperReports e o Jaspersoft ETL (é o mesmo que dizer Talend Open Studio) são módulos gratuitos e disponibilizados na versão gratuita/*community*, todos eles já foram analisados anteriormente. Os módulos que disponibilizam funcionalidades avançadas, como as listadas anteriormente, estão apenas disponíveis na versão comercial.



**Figura 25 – Arquitectura do Jaspersoft – versão completa**  
(fonte: <http://www.columnit.com/column-partners/jaspersoft/jaspersoft-business-intelligence.html>)

Os dois aspectos que o autor achou mais relevantes na versão comercial foi a API Visualize.js e a elegância dos gráficos e *dashboards* criados com esta versão. A API permite que os programadores consigam integrar, através da linguagem javascript, os documentos do Jaspersoft em aplicações *custom* (feitas p.e: em Microsoft .NET, Java, PHP, etc.) de forma muito subtil e fácil. Os gráficos e

*dashboards* são muito apelativos e simpáticos, permitem ainda que o utilizador interaja directamente em zonas de um gráfico ou *dashboard*. A Figura 26 mostra algumas *dashboards* que podem ser criadas com o Jaspersoft. Nos três *dashboards* da figura os dados são os mesmos, muda apenas o *layout*. Estes encontram-se para experimentação *live* no *website* da Jaspersoft. As *dashboards* mostram dados relativos a vendas em diversas zonas do globo, onde o utilizador ao carregar num país vê os restantes componentes da *dashboard* serem refrescados de maneira a mostrar os dados da zona seleccionada.



Figura 26 – Exemplo de *dashboards* criadas com o Jaspersoft – versão completa (fonte: <http://www.slideshare.net/jaspersoft/jaspersoft-dashboards-webinar-may-2013>)

O custo da licença da versão comercial não é público, é necessário contactar directamente o departamento comercial da TIBCO para obter informações, como a análise e utilização das edições ou *softwares* comerciais não são o objectivo deste trabalho, não se contactou a empresa para obter informações de *pricing*.

#### 3.6.2 Actuate BIRT

A ferramenta Actuate BIRT [83] nasceu do projecto BIRT Project. A empresa Actuate adquiriu os direitos deste último e passou a incluir o seu motor nas suas soluções. A operação de compra dos direitos do BIRT Project por parte da Actuate foi um processo muito semelhante à compra dos direitos do JasperReports por parte da TIBCO. O BIRT tem uma versão gratuita e duas versões comerciais. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	BIRT
<b>Download:</b>	<a href="http://www.actuate.com/resources/product-downloads/#designers">http://www.actuate.com/resources/product-downloads/#designers</a>
<b>Licença:</b>	EPL – versão gratuita
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2004
<b>Versão Actual:</b>	4.5.0
<b>Outros Aspectos:</b>	Ferramenta compatível com Windows e Unix.

O BIRT tem uma versão gratuita/*community*, mas, mais uma vez, tal como o Jaspersoft, a versão gratuita continua a ser o projecto *open source* adquirido, neste caso o BIRT Project. Tudo o que foi

dito em 3.3.2 aplica-se para o BIRT versão *gratuita/community*. Após a compra dos direitos a Actuate não adicionou nenhuma funcionalidade nova à versão *gratuita/community*, ou seja, o projecto gratuito da Actuate continua a ser o BIRT Project.

A versão Professional do Actuate BIRT chama-se iHub [83], disponibiliza aos utilizadores várias funcionalidades para lá dos *reports*, ao contrário da versão comercial do Jaspersoft, o iHub não foi alvo de experimentação. As funcionalidades que esta versão disponibiliza são muito idênticas às que o TIBCO Jaspersoft disponibiliza, são exemplo: *ad-hoc reports*, API de integração, *dashboards*, análise em memória, análises interactivas.

Como se pode constatar pelas funcionalidades anteriores, o Actuate BIRT e o TIBCO Jaspersoft são muito idênticos, nas suas edições comerciais, partilham basicamente as mesmas funcionalidades. A grande diferença é que a Actuate disponibiliza uma versão gratuita da versão comercial, mas existe uma restrição que não a torna viável para as organizações, o limite diário de *output* de dados. O Actuate F-Type (a versão *free* da versão comercial) é um produto bastante semelhante com o Actuate iHub (com menos funcionalidades avançadas, no entanto, todas as funcionalidades anteriormente listadas estão disponíveis no Actuate F-Type) mas tem um limite diário de utilização de dados restringido a 50MB, isto torna impossível a utilização a um nível empresarial, ficando restringido ao uso pessoal.

A Figura 27 mostra um exemplo de *dashboard* interactiva que pode ser criada utilizando o Actuate iHub (através da aplicação de gestão *web* fornecida com a licença no acto da compra, ou, alternativamente, através do editor gratuito Analytics Designer). A figura mostra uma *dashboard* interactiva, tal como a edição comercial da Jaspersoft, é possível criar *dashboards* com componentes interligados, que permitem ao utilizador filtrar dados da *dashboard* toda através da interacção com um componente específico da mesma.



**Figura 27 - Exemplo de *dashboards* criadas com o BIRT iHub**  
(fonte: [http://pt.appszoom.com/iphone\\_applications/business/birt-ihub-viewer-hd\\_gtiaz.html](http://pt.appszoom.com/iphone_applications/business/birt-ihub-viewer-hd_gtiaz.html))

Tal como o Jaspersoft, o custo da licença da versão comercial não é público. É necessário contactar directamente o departamento comercial da Actuate para obter informações. Mais uma vez, como a análise e utilização das edições ou *software*'s comerciais não são o objectivo deste trabalho, e não se

dedicou muito tempo à experimentação desta versão, não houve qualquer contacto com a empresa para obter informações de *pricing*.

#### 3.6.3 Jedox Palo

A ferramenta Jedox Palo [84][85] é uma ferramenta que surgiu em 2002 mantida pela empresa Jedox. O Palo tem uma versão gratuita e uma versão comercial. Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Jedox
<b>Download:</b>	<a href="http://sourceforge.net/projects/palo/">http://sourceforge.net/projects/palo/</a>
<b>Licença:</b>	General Public License – versão gratuita
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2002
<b>Versão Actual:</b>	5.1
<b>Outros Aspectos:</b>	Ferramenta compatível com Windows e Linux.

A versão comercial do Palo disponibiliza várias funcionalidades, são exemplo: *reporting*, *dashboarding*, OLAP, ETL, entre outras. Não se experimentou a versão de experimentação desta ferramenta, pois como já foi referido anteriormente, as versões comerciais não são o foco deste trabalho (apenas se abriu excepção pelo TIBCO Jaspersoft, por ser uma solução muito promissora). A versão gratuita do Palo é *open source* e disponibiliza essencialmente OLAP e ETL. Esta solução é constituída pelos seguintes componentes [84]:

- **OLAP Server:** servidor que permite correr as consultas OLAP;
- **ETL Server:** servidor responsável por correr os processos de ETL;
- **Palo Web:** aplicação *web-based* responsável por gerir as consultas OLAP e os processos ETL;
- **Palo para Excel:** *plugin* para o Microsoft Excel e o OpenOffice Calc que permitem fazer *reporting* e OLAP directamente nestas ferramentas.

O grande foco do Palo são as consultas OLAP, e o grande destaque e diferenciação desta *suite* é o Palo para Excel. Este *plugin* ao ser instalado acrescenta uma nova funcionalidade às ferramentas Microsoft Excel e OpenOffice Calc. Com este *plugin* as ferramentas ficam com um novo tabulador, e é disponibilizado um leque de funcionalidades como pivot tables e cubos OLAP. A Figura 28 mostra o Palo para Excel no Microsoft Excel.

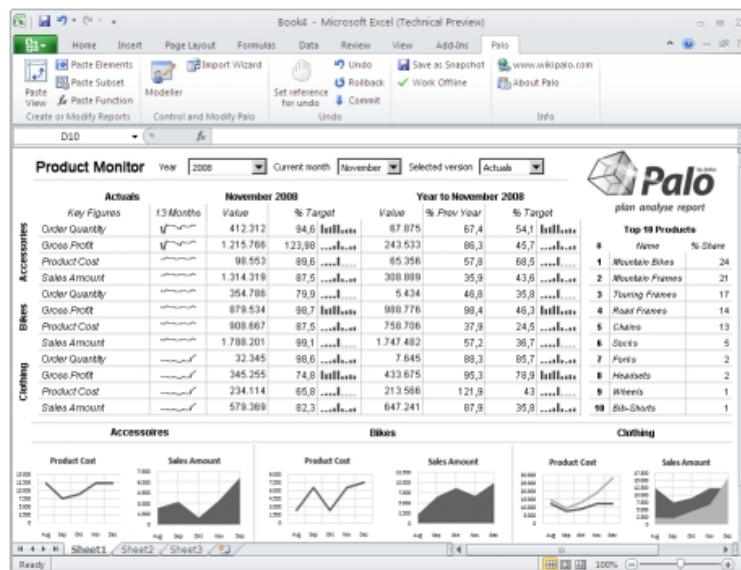


Figura 28 – *Plugim Palo para Excel a correr no Microsoft Excel*  
(fonte: <http://mybac.googlecode.com/svn/trunk/Quellen/Palo/info/PICS/>)

Palo para Excel tem as seguintes características [85]:

- Base de dados multidimensional em memória, isto é, os dados existentes no excel são colocados directamente numa base de dados;
- Consolidação de dados em *runtime* que permitem fazer operações de BI como por exemplo: *dashboarding* e *reporting*;
- Planeamento e análises direccionadas para a área financeira.

Estas características permitem a utilização do Palo para análises multidimensionais, em ferramentas conhecidas como folhas de cálculo. Os cálculos baseiam-se directamente no valor das células e não nos próprios registos, com esta característica é possível a criação de análises complexas de dados que apenas existem nas folhas de cálculo.

Como já foi referido, o que distingue o Palo da concorrência é o seu *plugim* que permite realizar análises, planeamento e *reporting* em *software* de folhas de cálculo. A maior limitação desta ferramenta é a licença do Palo para Excel. A utilização do Palo para Excel é gratuita ao nível do utilizador, isto é, numa organização,  $n$  utilizadores podem utilizar esta ferramenta nas suas “folhas de cálculo”, mas não é permitida a centralização de utilização, isto é, não é possível  $n$  utilizadores utilizarem esta ferramenta numa única máquina com dados centralizados, e isto é uma limitação muito grande. A *suite* Palo é uma solução interessante para quem deseja fazer ETL e OLAP, o seu *plugim* Palo para Excel permite a realizar de análises complexas directamente em folhas de cálculo, este *plugim* pode ser uma solução muito interessante para os departamentos financeiros (o próprio *plugim* é otimizado para a análise de dados com esta natureza).

### 3.6.4 Pentaho

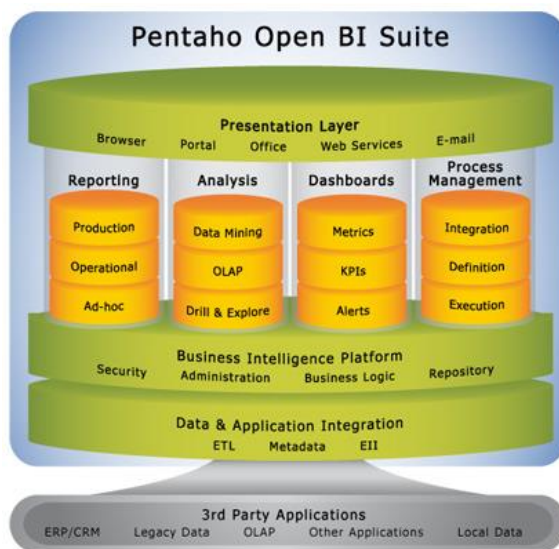
Pentaho [86][87][88] é uma das ferramentas de *business intelligence* mais divulgadas e conceituadas. Pentaho é uma ferramenta de apoio à decisão composta por dois tipos de versões, *community* que detém parte das funcionalidades totais da ferramenta sendo completamente gratuita (versão *open*

source) e uma versão *enterprise*, que contém todas as funcionalidades (versão comercial). Segue-se o cartão de visita desta ferramenta:

<b>Responsabilidade:</b>	Pentaho
<b>Download:</b>	<a href="http://community.pentaho.com/">http://community.pentaho.com/</a>
<b>Licença:</b>	General Public License – versão gratuita
<b>Tem Versão Paga:</b>	Sim
<b>Ano de Lançamento:</b>	2004
<b>Versão Actual:</b>	5.4.0.1
<b>Outros Aspectos:</b>	Ferramenta compatível com Windows, Unix e Linux.

A ferramenta Pentaho incorpora a ferramenta PDI para fazer ETL. Para o cruzamento de dados OLAP, incorpora as ferramentas Mondrian e Ramsetcube. Possui funcionalidades próprias para criação de relatórios de dados, inclui funcionalidades de *data mining* baseadas na aplicação Weka, e disponibiliza ainda funcionalidades relacionadas com a criação e visualização de *dashboards*. O Pentaho também permite fazer análises através de um mecanismo próprio, permitindo a criação e visualização de gráficos sendo possível a escolha de entre diversos tipos de gráficos existentes na plataforma. É possível a criação de relatórios, relatórios de análise e dashboards. Os relatórios podem ser visualizados e exportados nos formatos de html, pdf, xls, csv, rtf e plain text.

A linguagem MDX é utilizada para as consultas à base de dados integrada na ferramenta, sendo também ela a tecnologia que sustenta as funcionalidades de OLAP da aplicação, as consultas podem ser exportadas em ficheiros pdf ou xls. A Figura 29 mostra a *stack* que forma a solução Pentaho.



**Figura 29 – Componentes que formam o Pentaho**  
(fonte: <http://www.wexer.ro/en/pentaho-bi-suite>)

A plataforma Pentaho é composta de aplicações para criação de soluções de BI. Entre os módulos destacam-se [87]:

- **BI Server:** *front-end* de interação e gestão para o utilizador final. O servidor pode ser utilizado de dois modos diferentes, através do Pentaho User Console (PUC) e/ou do Pentaho

Administration User (PAC). Enquanto a primeira é uma aplicação do tipo *console* a segunda é uma aplicação *web-based*;

- **Report Designer:** ferramenta utilizada para criar *reports*. Quando for necessário criar relatórios mais interativos e elaborados esta ferramenta é a mais adequada, pois fornece mais recursos que a ferramenta de geração de relatórios ad-hoc;
- **Design Studio:** esta é uma ferramenta que é baseada no IDE Eclipse (API para desenvolvimento). Podem-se criar documentos como por exemplo: *dashboards*, *gráficos*, *KPI's*, etc.;
- **Aggregation Designer:** ferramenta gráfica que ajuda a melhorar a eficiência do cubo Mondrian, criando tabelas agregadas;
- **Metadata Editor:** ferramenta que faz os mapeamentos com as bases de dados de modo a que os utilizadores possam utilizar os mesmos para criarem relatórios via web. O seu uso não é obrigatório, mas a criação de metadados facilita a criação de *reports* e outros documentos;
- **Pentaho Data Integration:** como já foi referido anteriormente, o PDI é uma ferramenta de ETL, todas as suas características podem ser revistas em 3.2.2;
- **Pentaho Schema Workbench:** ferramenta gráfica que permite a criação de esquemas OLAP para análise. Para fazer análise de cubos OLAP é necessário criar os esquemas nesta ferramenta .

A Figura 30 mostra uma *dashboard* que foi criada pela ferramenta Pentaho, sendo importante referir que esta é retirada de uma *demo* que a própria empresa utiliza para divulgar a ferramenta.



Figura 30 – Exemplo de dashboard criada com o Pentaho  
(fonte: <http://sourceforge.net/projects/pentaho/>)

Como se pode ver pela figura anterior a ferramenta permite a utilização de gráficos, KPI's, tabelas e filtros para fazer análises de negócio.

### 3.6.5 SpagoBI

SpagoBI [89][90] é uma aplicação desenvolvida pela Engineering Group, fundada em 2006 e suportada por uma comunidade *open source*. A ferramenta SpagoBI permite aos utilizadores usufruir



As propriedades OLAP disponibilizadas pela ferramenta permitem uma análise multidimensional efectuada com recurso às perspectivas de *drill-down*, *drill-across* e *slice and dice*. Os processos de OLAP são suportados pelos motores OLAP JPivot/Mondrian, JPalo/Mondrian e JPivot/XMLA Server.

As propriedades gráficas têm como suporte a ferramenta JFreeChart, que permite o desenho de gráficos de diferentes tipos, barras, circulares, histogramas, etc.

A ferramenta disponibiliza mecanismos de *dashboarding*, permite a visualização de gráficos no formato swf, exibindo KPI's, sendo estes KPI's são exibidos em tempo real. A ferramenta permite a criação, gestão e visualização de KPI's, permitindo aplicar diferentes especificações, entre elas, regras de cálculo, limites e regras de alarme, que condicionarão a ilustração do gráfico final.

A SpagoBI detém também uma funcionalidade para a elaboração de *cockpits* que permite a junção de diferentes documentos em apenas um, isto é, permite a elaboração de um documento capaz de conter elementos de OLAP, *dashboards* e *data mining*, entre outros.

A ferramenta disponibiliza mecanismos geográficos, que permitem interligar dados em tempo real com dados provenientes de uma *data warehouse*. Esta funcionalidade é suportada por duas aplicações, GEO e GIS.

A vertente de *data mining* nesta ferramenta é suportado pela aplicação Weka. Esta permite a extração de padrões e informação relevante de uma quantidade de dados considerável.

A ferramenta fornece um motor próprio de consultas, *Query by Example* (QBE), que permite que as consultas sejam definidas pelos utilizadores de forma gráfica. Esta funcionalidade permite ainda a elaboração de consultas, verificação de resultados, exportação de dados, guardar consultas e criação de modelos de relatórios.

A opção *Smart Filter* permite criar formulários de consulta simples, utilizando a opção de filtragem de dados de forma a poder seleccionar informação mais válida e específica.

As funcionalidades de ETL do SpagoBI integram a aplicação TOS.

A aplicação permite a publicação de documentos nos formatos do Microsoft Office e Open Office.

A ferramenta é constituída por vários módulos, o SpagoBI Studio (permite criar e modificar documentos de análise como relatórios, gráficos etc.), o SpagoBI Meta (aplicação mais ligada à área de metadados) e o SpagoBI SDK (serviço que agrupa as funcionalidades da aplicação suportadas pelo servidor).

SpagoBI é um pacote de funcionalidades de BI de grande dimensão. O seu conjunto de funcionalidades torna-a uma plataforma composta e com potencial. Face a outras ferramentas apresenta um grande número de funcionalidades e disponibiliza-as numa única versão sem quaisquer custos. A incapacidade de realizar consultas *ad-hoc* é o ponto frágil deste pacote de funcionalidades face a outros existentes no mercado.

A Figura 32 **Figura 30** mostra uma *dashboard* que foi criada pela ferramenta SpagoBI, ilustrando algumas propriedades de BI possíveis de execução por parte da ferramenta SpagoBI, nomeadamente a criação de gráficos de barras em 3D, tabelas de dados e gráficos de progressão em tempo real.



**Figura 32 – Exemplo de dashboard criado com o SpagoBI**  
(fonte: <http://www.spagobi.org/2014/11/spagobi-for-pagoda-the-open-lab-governance-dashboard/>)

### 3.7 Comparação entre as Ferramentas

Depois de analisadas e experimentadas várias ferramentas, de diversos tipos e para diferentes áreas, há que apresentar as conclusões. Em primeiro lugar serão abordadas as ferramentas que não são *suite*. Apesar de já terem sido comparadas anteriormente, serão apresentadas as principais conclusões. Depois será então apresentada uma comparação e análise crítica às ferramentas do tipo *suite*.

#### 3.7.1 Ferramentas Não Suite

Da secção 3.2 à 3.5 foram apresentadas várias ferramentas que são consideradas não *suite*, assim denominadas por disponibilizarem um conjunto de funcionalidades específicas de uma área de BI em particular. Ao longo das várias secções foram feitas comparações entre ferramentas da mesma área. Segue-se uma análise comparativa resumida das mesmas.

No que toca às ferramentas de ETL, Talend Open Studio (3.2.1) e Pentaho Data Integration (3.2.2), conclui-se o seguinte [59]:

- Ambas são muito parecidas no que toca ao modo de desenvolvimento dos processos de ETL;
- A curto prazo, o PDI é mais fácil de aprender. No entanto, a *interface* mais organizada do TOS faz com que esta ferramenta, após a fase inicial de aprendizagem, se torne mais simples de usar;
- O TOS é mais amigo do programador, pois o *debug* é poderoso, ao contrário do PDI que é muito limitado e básico;

- O TOS promove a reutilização de código. Os utilizadores podem criar sub-processos que podem ser reutilizados posteriormente noutros. O PDI não permite a criação de sub-processos.

O TOS e o PDI revelaram-se boas ferramentas para fazer ETL. Sendo qualquer delas uma opção viável para as organizações. O número de funcionalidades e acesso a fonte de dados é praticamente o mesmo. Neste projecto o TOS foi o preferido, pois a sua interface é mais usável e para o programador é mais simpática devido ao poderoso *debug* e à criação de sub-processos. É importante referir que ambas as ferramentas são utilizadas noutras *suites*: no caso do PDI é utilizada na solução Pentaho e o TOS é utilizado no SpagoBI e no TIBCO Jaspersoft.

No que toca às ferramentas de *reporting*, JasperReports (3.3.1) e BIRT Project (3.3.2), concluiu-se o seguinte [66]:

- A qualidade de imagem dos *reports* do JasperReports é superior pois são *pixel-based*. No entanto, pelos testes efectuados, esta diferença em aplicações *web* não é notória;
- O IDE de desenvolvimento do JasperReports tem muitos *bugs*, tendo acontecido algumas vezes que o programa *crashava*, enquanto se editavam relatórios. O IDE do BIRT é mais estável e simples de utilizar;
- O JasperReports permite criar sub-relatórios, tornando a tarefa de criação de relatórios complexos mais fácil. O controlo total sobre os estilos do relatório (através de CSS) é também uma mais valia que torna a criação de relatórios mais poderosa do que no BIRT;
- Se for necessário utilizar múltiplas fontes de dados para o mesmo relatório, o JasperReports pode ser uma dor de cabeça, pois os subrelatórios só podem utilizar a fonte de dados do relatório principal (é possível minimizar isto através de *views* que retornem todos os dados importantes para o relatório). A utilização de múltiplas fontes de dados no BIRT é feita de uma forma muito simples e fácil.

Em conclusão pode dizer-se que as duas ferramentas são poderosíssimas, são, sem qualquer dúvida, opções viáveis para fazer *reporting* nas organizações. A escolha de uma delas depende muito dos critérios que os utilizadores utilizem para as comparar. Se o controlo total dos *reports* (através de subrelatórios, estilos e *design*) é a prioridade, então o JasperReports é a solução a escolher. O autor preferiu o BIRT Project devido ao facto do IDE ser estável e simples de utilizar, a fácil utilização de múltiplas fontes de dados também foi um factor importante na preferência. O BIRT Project só falha por não permitir criar sub-relatórios.

No que toca às ferramentas de OLAP, Mondrian (3.4.1) e JPivot (**Error! Reference source not found.**), concluiu-se o seguinte:

- As duas ferramentas não devem ser comparadas por terem propósitos diferentes. O Mondrian é um servidor de dados multidimensionais e o JPivot é uma API para mostrar e manipular dados;
- Ambas as ferramentas têm boas *performances* no que toca a velocidade de processamento dos dados e operações sobre os mesmos;
- O JPivot, apesar de ter uma *interface* antiquada, permite a realização de várias operações OLAP de forma sublime, permite ainda a exportação dos dados para diversos formatos.

Através da experimentação efectuada pelo autor, o mesmo chega à conclusão que as duas devem complementar-se de modo a que os utilizadores possam ter uma solução OLAP 2 em 1, ou seja, servidor e API de dados ao mesmo tempo. É importante referir que estas duas ferramentas são utilizadas pelo Pentaho e SpagoBI nas suas componentes de OLAP.

Já no que concerne toca às ferramentas de *data mining*, Weka (3.5.1) e Rapid Miner (3.5.2), concluiu-se o seguinte [79]:

- O Weka torna a comparação de diferentes algoritmos para o mesmo problema a analisar, uma tarefa fácil;
- O Rapid Miner tem uma curva de aprendizagem menor, pois existe muita literatura e ajuda *online*;
- O Rapid Miner tem um corrector de código automático que ajuda os utilizadores no momento de criação de *flows* de dados;
- O Rapid Miner disponibiliza outras funcionalidades para lá de *data mining* (*dashboarding*, *reporting*, ETL, entre outras).

Ambas as ferramentas são poderosas no que toca às funcionalidades de *data mining* disponibilizadas. O Rapid Miner não faz só *data mining*, disponibilizando outras funcionalidades. O Weka torna a comparação de diferentes algoritmos, para a mesma análise, uma tarefa fácil. Os utilizadores que pretendem apenas fazer *data mining* sem outras distrações, o Weka é a ferramenta a escolher. Para os utilizadores, menos experientes nesta área, podem ver no Rapid Miner uma solução aliciante pois existe muita documentação para dar os primeiros passos, para lá deste aspecto, esta ferramenta disponibiliza outras funcionalidades.

Em suma, todas as ferramentas analisadas e experimentadas, na opinião do autor, são soluções viáveis para as organizações. Todas as ferramentas não-*suite* podem ser usadas para uso comercial, no entanto os utilizadores devem ter cuidado na escolha das versões gratuitas, pois existem outras versões pagas que são consideradas comerciais. Algumas destas ferramentas são usadas nas soluções do tipo *suite*, têm por isso provas dadas quanto à sua qualidade.

#### 3.7.2 Ferramentas Suite

Da secção 3.6.1 à 3.6.5 foram apresentadas várias ferramentas que são consideradas *suite*, sendo consideradas como tal por disponibilizarem um conjunto de funcionalidades alargadas que respondem a várias áreas de BI. Segue-se uma análise comparativa.

O primeiro ponto a analisar, neste tipo de ferramentas, é o conjunto de funcionalidades que oferecem. É importante referir que as comparações são feitas entre as versões gratuitas das ferramentas. Foi referido que algumas destas ferramentas têm uma versão comercial, mas este tipo de versões estão fora do âmbito deste documento/estudo.

A Tabela 4 mostra as funcionalidades que cada ferramenta disponibiliza aos utilizadores finais [91][92].

Tabela 4 – Comparação entre as funcionalidades disponibilizadas

Funcionalidades	TIBCO Jaspersoft	Actuate BIRT	SpagoBI	Pentaho	Jedox Palo
Reporting	✓	✓	✓	✓	✓
Ad-Hoc Reporting	X	X	X	✓	X
Dashboards	X	X	✓	✓	✓
Data Mining	X	X	✓	✓	✓
ETL	✓	X	✓	✓	✓
Exportação de Dados	✓	✓	✓	✓	✓
Gráficos	X	X	✓	✓	X
KPI's	X	X	✓	✓	✓
OLAP	X	X	✓	✓	✓

Pela análise da tabela anterior conclui-se que as ferramentas TIBCO Jaspersoft e Actuate BIRT, nas suas versões gratuitas, são as *suites* que menos funcionalidades disponibilizam. Já as ferramentas Pentaho, SpagoBI e Jedox Palo são as *suites* que disponibilizam mais funcionalidades.

O TIBCO Jaspersoft e o Actuate BIRT disponibilizam todas as funcionalidades nas suas versões comerciais. No entanto, as suas versões gratuitas são basicamente os projectos *open source* de *reporting* JasperReports e BIRT Project. Estes dois projectos *open source* foram adquiridos por empresas que, posteriormente, adicionaram outras funcionalidades e disponibilizaram estas em versões comerciais, à excepção do TIBCO Jaspersoft que adicionou à versão gratuita a funcionalidade de ETL (através do Talend Open Studio). Conclui-se então, de forma muito clara, que as versões gratuitas do TIBCO Jaspersoft e do Actuate BIRT são as versões *open source* dos projectos iniciais adquiridos, com a excepção já referida do TIBCO Jaspersoft que adicionou a funcionalidade de ETL.

O Jedox Palo é uma ferramenta que se destaca pelo seu *plugin* que torna possível fazer OLAP, *reporting*, KPI's e *data mining* de forma directa em folhas de cálculo, através das ferramentas Microsoft Excel e OpenOffice Calc. O Jedox Palo também permite fazer ETL. Esta ferramenta revela uma grande limitação, a licença não permite partilhar dados entre utilizadores diferentes, isto é, não é possível dois utilizadores fazerem análises a dados a correr num único servidor. Na versão gratuita desta ferramenta é necessário que cada utilizador instale, separadamente, o servidor das análises e o *plugin* para fazer análises no *software* de cálculo. As funcionalidades disponibilizadas, à excepção do ETL, são feitas sempre nas folhas de cálculo, logo as exportações de dados, gráficos, *reports*, *dashboards* e outras funcionalidades dependem também das aplicações de cálculo onde o *plugin* é corrido.

Continuando a análise, pode observar-se que o Pentaho é a ferramenta que disponibiliza todas as funcionalidades enquanto que o SpagoBI apenas não disponibiliza *ad-hoc reporting*. Estas duas ferramentas são as que apresentam melhores argumentos para as organizações que pretendem ter uma única ferramenta a tratar das operações de BI. As funcionalidades de ETL, *reporting*, *dashboarding*, *data mining* e OLAP são as mais importantes no contexto de BI, dependendo sempre das necessidades da organização, em particular, sendo estas as mais procuradas, e o Pentaho e o SpagoBI disponibilizam todas elas.

Outro aspecto que é importante analisar é a compatibilidade entre sistemas operativos diferentes. Existem organizações que por motivos de infraestruturas e custos das mesmas, analisam os sistemas operativos em que as soluções, das mais diversas áreas, podem correr. A maioria das organizações utiliza sistemas Windows, mas isso não significa que os produtos finais desenvolvidos não sejam corridos em Unix ou Linux, com ênfase especial para o Linux por ser um sistema operativo gratuito e *open source*. A Tabela 5 apresenta a compatibilidade das ferramentas nos principais sistemas operativos [93][94].

**Tabela 5 – Comparação entre os sistemas operativos suportados**

<b>Sistema Operativo</b>	TIBCO Jaspersoft	Actuate BIRT	SpagoBI	Pentaho	Jedox Palo
Linux	✓	✗	✓	✓	✓
Unix	✓	✓	✓	✓	✗
Windows	✓	✓	✓	✓	✓

A ferramenta Actuate BIRT pode ser excluída pelas organizações por não poder ser corrida no sistema Linux, já o Jedox Palo levantará problemas a organizações que desenvolvam e/ou disponibilizem soluções em sistemas Unix. O TIBCO Jaspersoft, o Pentaho e o SpagoBI podem correr em qualquer dos sistemas operativos analisados.

Como já foi referido anteriormente, existem ferramentas que para lá da versão gratuita, disponibilizam versões comerciais, ou seja, versões pagas. São exemplo disso as *suites* TIBCO Jaspersoft, Actuate BIRT, Pentaho e Jedox Palo. Outro aspecto relevante para analisar são as versões disponibilizadas e as funcionalidades que podem correr em dispositivos móveis (telemóveis, tablets, etc.). Nos dias de hoje existem muitas aplicações, nas mais diversas áreas, que são optimizadas para este tipo de dispositivos. As aplicações de BI começam a ter esta portabilidade em conta [94].

**Tabela 6 - Comparação entre as versões disponíveis**

<b>Versão</b>	TIBCO Jaspersoft	Actuate BIRT	SpagoBI	Pentaho	Jedox Palo
Community	✓	✓	✓	✓	✓
Enterprise	✓	✓	✗	✓	✓

Mobile	✓ **	✓ **	✓	✓ **	✓ **
--------	------	------	---	------	------

\*\* - Apenas na versão paga

Como se pode constatar pela análise da tabela anterior, o SpagoBI é a única ferramenta 100% *open source* pois não tem nenhuma versão comercial. As versões de *mobile* só estão disponíveis nas versões pagas de todas as ferramentas, à exceção do SpagoBI que não tem este tipo de versão. As organizações que tenham como requisito uma ferramenta gratuita com a componente *mobile* também gratuita têm apenas uma opção, o SpagoBI.

### 3.8 Ferramenta Escolhida

Neste capítulo foram analisadas, demonstradas e comparadas todas as ferramentas passíveis de utilização no contexto do presente trabalho. Desde a área de *reporting*, passando pela área de OLAP e acabando em *suites* que disponibilizam um conjunto alargado de funcionalidades. Consegue-se perceber que existem várias opções para as empresas criarem aplicações de BI com ferramentas gratuitas.

Quando no início deste trabalho, se encetou a pesquisa, identificação e teste (utilização) das várias ferramentas, a organização para onde foi criado o caso de teste (que será demonstrado no capítulo seguinte), reuniu um conjunto de características desejáveis para a(s) ferramenta(s) a escolher, são elas:

- Disponibilização das seguintes funcionalidades: ETL, *reporting* e *ad-hoc reporting* (e exportação de dados em vários formatos diferentes, sendo os formatos de pdf e excel obrigatórios), *dashboarding* e gráficos;
- O sistema operativo preferencial é o Linux, no entanto, caso não seja possível, o sistema Windows deve ser a opção;
- Preferência por ferramentas do tipo *suite*, no entanto, tal como o caso anterior, este requisito não é eliminatório;
- As ferramenta(s) escolhida(s) não deve(m) apresentar uma curva de aprendizagem muito elevada para a equipa de BI;
- Deve existir documentação suficiente e com qualidade de modo a ajudar os *developers* e outros envolvidos a resolver problemas técnicos;
- Preferência por ferramentas que tenham suporte técnico especializado (pago ou não).

O primeiro ponto analisado, por ser um requisito obrigatório, constitui-se como o conjunto de funcionalidades obrigatórias, da análise relativa às características das várias ferramentas experimentadas, pode concluir-se que:

- As ferramentas TOS, PDI, Pentaho, Jedox Palo e TIBCO Jaspersoft suportam a funcionalidade de ETL;
- As ferramentas JasperReports, BIRT Project, TIBCO Jasperfot, Actuate BIRT, Pentaho, SpagoBI e Jedox Palo suportam a funcionalidade de *reporting*;

- Apenas a ferramenta Pentaho suporta a funcionalidade de *ad-hoc reporting* (as ferramentas TIBCO Jaspersoft e Actuate BIRT suportam apenas nas suas versões pagas);
- As ferramentas Pentaho, SpagoBI e Jaspersoft suportam a funcionalidade de *dashoarding* (e respectiva criação de gráficos independentes).

Através da análise anterior pode-se concluir que apenas o Pentaho suporta todas as funcionalidades obrigatórias, pois é a única ferramenta que permite fazer *ad-hoc reporting*. O SpagoBI apesar de não disponibilizar o desenvolvimento de *report's ad-hoc* puros, tem duas funcionalidades que servem a mesma intenção, são elas o QBE e o SmartFilter. O QBE permite que os utilizadores criem as suas próprias consultas (através dos modelos de dados disponibilizados) através de *drag'n'drop* de componentes (tabelas, gráficos, etc.) e o SmartFilter permite que os utilizadores filtrem dados de uma fonte escolhida. Apesar destas últimas duas funcionalidades não serem um *ad-hoc reporting* puro (onde os utilizadores editam e manipulam os seus próprios relatórios), se forem bem utilizadas e os modelos de dados disponibilizados forem flexíveis e robustos, os utilizadores continuam a ter a capacidade de construir as suas próprias consultas. Resumindo, o Pentaho e o SpagoBI são as ferramentas que dão resposta a todas as funcionalidades exigidas.

O segundo ponto analisado, foi a compatibilidade das ferramentas para os sistemas operativos Linux e Windows. Desta análise conclui-se que todas as ferramentas, não *suite* e *suite*, podem ser executadas nos dois sistemas, excepto o Actuate BIRT.

Até ao momento, pela análise dos dois pontos anteriores, chega-se à conclusão que apenas duas ferramentas respondem aos requisitos analisados, são elas o Pentaho e o SpagoBI.

O terceiro ponto analisado foi a preferência por ferramentas do tipo *suite*. Como foi referido no parágrafo anterior, apenas o Pentaho e o SpagoBI respondem aos primeiros dois pontos analisados, como ambas são do tipo *suite*, então a preferência por este tipo de ferramentas é verificado pois ambas são *suite* com um conjunto alargado de funcionalidades.

O quarto, quinto e sexto pontos foram analisados e explorados aos mesmo tempo, ou seja, o nível de aprendizagem, documentação e suporte técnico foi avaliado para as ferramentas Pentaho e SpagoBI. Estas análises foram as últimas a serem feitas e como apenas o Pentaho e o SpagoBI cumprem os requisitos até ao momento, então foram estas duas que foram comparadas nestes três pontos. As Tabela 7, Tabela 8 e Tabela 9 demonstram as comparações feitas nestes três pontos.

**Tabela 7 – Comparação quanto à curva de aprendizagem**

<b>Curva de Aprendizagem</b>	Pentaho	SpagoBI
Elevada		
Consideravelmente elevada	●	
Aceitável		

Baixa		●
-------	--	---

**Tabela 8 – Comparação do número e qualidade de documentação existente**

<b>Tipo de Documentação Existente</b>	Pentaho	SpagoBI
Documentação oficial gratuita	3	2
Documentação oficial paga	3	4
Documentação digital gratuita	5	4
Documentação digital paga	5	3
Tutoriais <i>online</i>	5	4

- 1 – Nenhuma encontrada
- 2 – Pouca e com qualidade insuficiente
- 3 – Número considerável e com qualidade suficiente
- 4 – Número considerável e com boa qualidade
- 5 – Muita e com boa qualidade

**Tabela 9 – Comparação do suporte técnico**

<b>Suporte Técnico</b>	Pentaho	SpagoBI
Apenas fóruns <i>online</i> para a versão gratuita	●	
Consultoria e suporte técnico gratuito		
Consultoria e suporte técnico pago		●
Consultoria e suporte técnico pago para a	●	

versão comercial		
------------------	--	--

De acordo com as comparações efectuadas conclui-se o seguinte:

- O SpagoBI é mais fácil de aprender e utilizar que o Pentaho;
- O Pentaho oferece mais e melhor documentação no geral;
- O SpagoBI tem suporte técnico, mas pago, para a sua versão gratuita. O Pentaho apenas tem suporte, para a sua versão gratuita, através dos *forums online*.

O SpagoBI ganha em dois dos três pontos analisados, ficando apenas a perder quanto à quantidade e qualidade da documentação existente. No entanto, a existência de suporte e consultoria técnica especializada, apesar de paga, colmata o problema da documentação. O Pentaho, por seu lado, revelou-se complicado de aprender e utilizar sendo o SpagoBI mais fácil e intuitivo. O pior aspecto do Pentaho é a inexistência de suporte especializado para a sua versão gratuita, isto pode fazer com que os responsáveis pelas equipas de desenvolvimento e outros associados possam ter problemas que poderão demorar muito tempo a resolver, ou no pior dos casos, ficarem por resolver.

Depois de todo o processo de pesquisa, identificação, experimentação, comparação e análise crítica, foi recomendado à instituição o uso da ferramenta SpagoBI, por todos os motivos referidos anteriormente, esta ferramenta vai ao encontro das necessidades da organização.

## 4. Problema para Aplicabilidade Prática

No capítulo anterior foi apresentada a investigação realizada sobre as possíveis ferramentas gratuitas para o desenvolvimento de aplicações de BI. No fim do capítulo, após várias comparações e verificações dos requisitos iniciais concluiu-se que a ferramenta SpagoBI era a mais adequada para a organização.

Este trabalho, para lá da investigação e experimentação de ferramentas de BI gratuitas, tem um cunho bastante prático. Dizer que a ferramenta *a* ou a ferramenta *b* são as melhores é, no mínimo, vago. É preciso provas práticas para se verificar tal. O objectivo do trabalho é provar que existem ferramentas gratuitas capazes de permitir o desenvolvimento de aplicações BI e que conseguem produzir soluções com qualidade capazes de competir com outras soluções comerciais, e portanto, pagas. No capítulo anterior foi provado que existem, efectivamente, várias ferramentas gratuitas, com capacidade de criar soluções de BI gratuitamente. Falta verificar a qualidade das soluções produzidas. Este capítulo tem como objectivo apresentar o problema prático na qual vai ser implementada uma solução de BI com a ferramenta SpagoBI, permitindo assim, avaliar a ferramenta quanto a este último requisito.

### 4.1 Sistema de Controlo e Acompanhamento de Projectos Internos

O sistema que vai constituir o caso de aplicação da ferramenta SpagoBI, foi desenvolvido pela organização Pessoas e Processos para um cliente internacional. Alguns detalhes do projecto serão ocultados por motivos de direitos de autor e confidencialidade para com a organização e o cliente. Apesar da confidencialidade que será respeitada, a informação que será apresentada é a suficiente para os leitores deste documento perceberem o sistema, os seus objectivos e os seus módulos principais.

O objectivo da aplicação é auxiliar a organização cliente a fazer a gestão dos seus projectos internos, contratos e facturação, sendo . A aplicação é composta pelos seguintes módulos funcionais:

- **Sistema de informação para gestão de contratos:** conjunto de funcionalidades para gestão detalhada de contratos que a organização queira controlar. De forma simples, será a

ferramenta que permite editar e registar as características dos contratos e acompanhar todo o ciclo de vida de um contrato, esteja ou não associado a projectos. Permite criar ou apenas assegurar o suporte administrativo aos contratos (estejam ou não associados a projectos ou actividades), permitindo o registo e controlo dos respectivos planos de pagamento;

- **Sistema de informação para controlo e acompanhamento de projectos:** conjunto de funcionalidades para gestão detalhada de projectos e actividades correntes que necessitem de um controlo físico e financeiro da sua execução;
- **Módulo de gestão de *backoffice*:** conjunto de funcionalidades para a gestão da informação base (utilizadores, departamentos, classificadores, etc.) e a parametrização do sistema (conjunto de valores que influenciarão o comportamento automático do sistema de informação);
- **Portal de Conhecimento:** conjunto de funcionalidades que irão integrar a informação dos diversos sistemas sob a forma de *dashboards* executivos. Inclui funcionalidades de exploração livre da informação de projectos e contratos na procura de evidências de negócio (*business analysis*) e outras funcionalidades que permitem o desenvolvimento de conteúdos específicos e necessários à gestão da organização como os *reports* ou *dashboards*.

#### 4.1.1 Sistema de Informação para Gestão de Contratos

O módulo contratos é constituído por um conjunto de funcionalidades que permitem assegurar a gestão detalhada de contratos ao longo do seu ciclo de vida. Encontra-se organizado por contratos, facturas, documentos de autorização e listagens.

Os contratos classificam-se como vigentes ou arquivados de acordo com o estado em que encontram no decorrer do seu fluxo. Consideram-se contratos vigentes os que se encontram em caracterização, validação, à espera de execução, em execução ou em alteração. Os contratos que se encontram nos estados cancelado, suspenso, rescindido ou concluído consideram-se arquivados. A descrição de cada estado é desnecessária. Interessa saber que um contrato é vigente ou arquivado, dependendo do estado actual em que se encontra. A Figura 33 mostra o *flow* da transição de estados que um contrato pode sofrer ao longo do seu ciclo de vida.

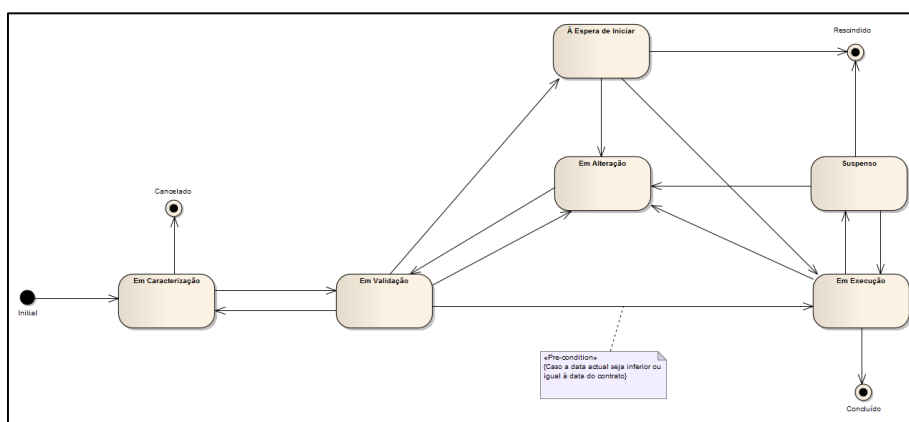


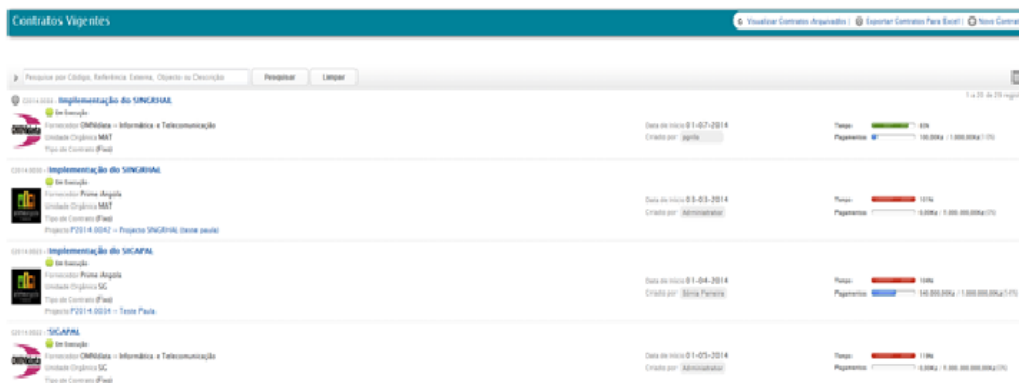
Figura 33 – Ciclo de vida de um contrato

A componente para a gestão de contratos disponibiliza aos utilizadores as seguintes páginas principais:

## 4 – Problema para Aplicabilidade Prática

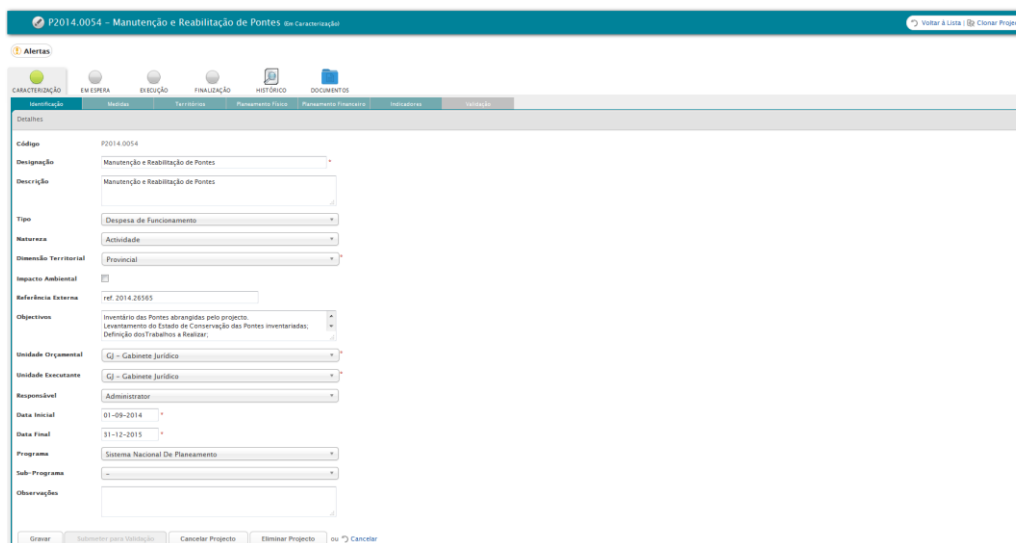
- **Listagem de contratos:** lista de todos os contratos (vigentes ou arquivados) existentes e activos no sistema. Nesta página é possível filtrar os contratos por diversos campos e ainda exportar a listagem para excel (de acordo com os filtros aplicados);
- **Páginas de edição de contratos:** páginas que permitem editar as informações dos contratos. A informação a inserir está agrupada em vários separadores e em vários tabuladores.

As Figura 34, Figura 35 e Figura 36 mostram a listagem de contratos e duas páginas de edição dos contratos.



Código	Designação	Data de Início	Estado	Pagamentos
0014.0054	Implementação do SINGEMIA Instituto ONDData - Informática e Telecomunicação Unidade Orgânica MAT Tipo de Contrato (F)Aut	01-07-2014 Criado por: JGFR	Ativo	100.000€ / 1.000.000€ (10%)
0014.0055	Implementação do SINGEMIA Instituto Ponta Aberta Unidade Orgânica MAT Tipo de Contrato (F)Aut Projecto P2014.0042 - Projecto SINGEMIA (Sempre ativo)	03-03-2014 Criado por: Administrador	Ativo	0.000€ / 0.000.000€ (0%)
0014.0053	Implementação do SIGAFAL Instituto Ponta Aberta Unidade Orgânica ISC Tipo de Contrato (F)Aut Projecto P2014.0034 - Teixo Paulo	01-04-2014 Criado por: Maria Patricia	Ativo	140.000.000€ / 1.000.000.000€ (14%)
0014.0052	SIGAFAL Instituto ONDData - Informática e Telecomunicação Unidade Orgânica ISC Tipo de Contrato (F)Aut	01-07-2014 Criado por: Administrador	Ativo	0.000€ / 0.000.000.000€ (0%)

Figura 34 – Página de listagens de contratos



**P2014.0054 - Manutenção e Reabilitação de Pontes**

**Alertas:**

**Caracterização:** EM ESPERA, EXECUÇÃO, FINALIZAÇÃO, HISTÓRICO, DOCUMENTOS

**Identificação:** Detalhes

**Código:** P2014.0054

**Designação:** Manutenção e Reabilitação de Pontes

**Descrição:** Manutenção e Reabilitação de Pontes

**Tipo:** Despesa de Funcionamento

**Natureza:** Actividade

**Dimensão Territorial:** Provincial

**Impacto Ambiental:**

**Referência Externa:** ref. 2014.2555

**Objectivos:** Inventário das Pontes abrangidas pelo projecto. Levantamento do Estado de Conservação das Pontes inventariadas. Definição dos Trabalhos a Realizar.

**Unidade Orçamental:** GJ - Gabinete Jurídico

**Unidade Executante:** GJ - Gabinete Jurídico

**Responsável:** Administrador

**Data Inicial:** 01-09-2014

**Data Final:** 31-12-2015

**Programa:** Sistema Nacional De Planeamento

**Sub-Programa:** -

**Observações:**

Gravar | Submeter para Validação | Cancelar Projecto | Eliminar Projecto ou Cancelar

Figura 35 – Página de edição de contrato (1)

#### 4 – Problema para Aplicabilidade Prática

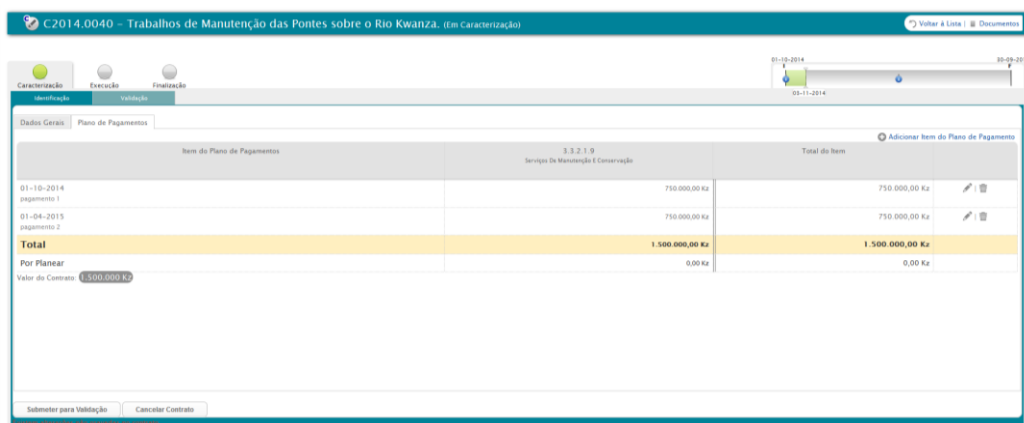


Figura 36 – Página de edição de contrato (2)

Este módulo é ainda responsável pela gestão de facturas. Uma factura possui um ciclo de vida e pode ser associada a um contrato. Para criar uma factura é necessário preencher diversas informações, este preenchimento, tal como nos contratos é feito através de vários separadores e tabuladores. A Figura 37 mostra o fluxo de vida de uma factura.

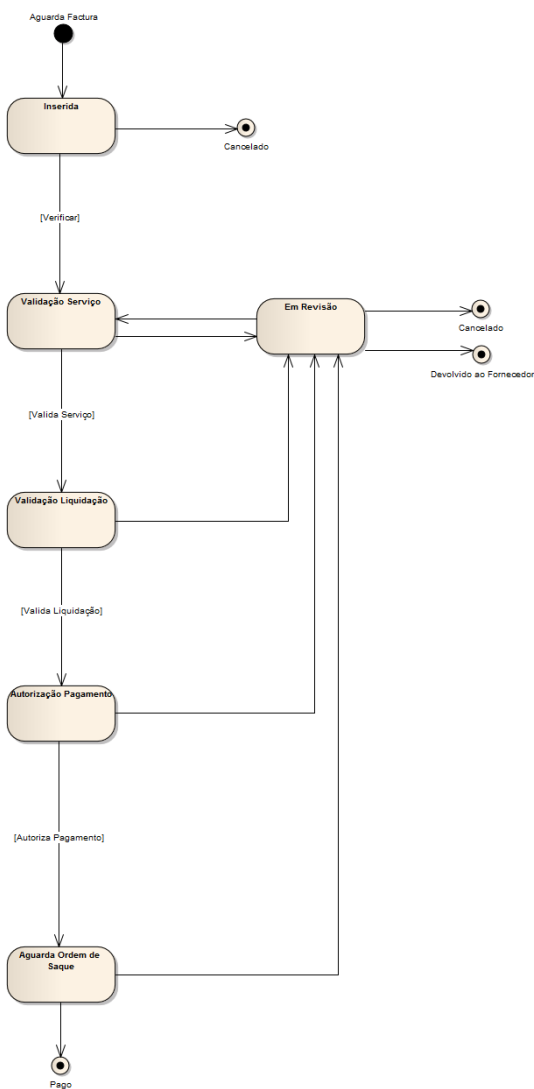


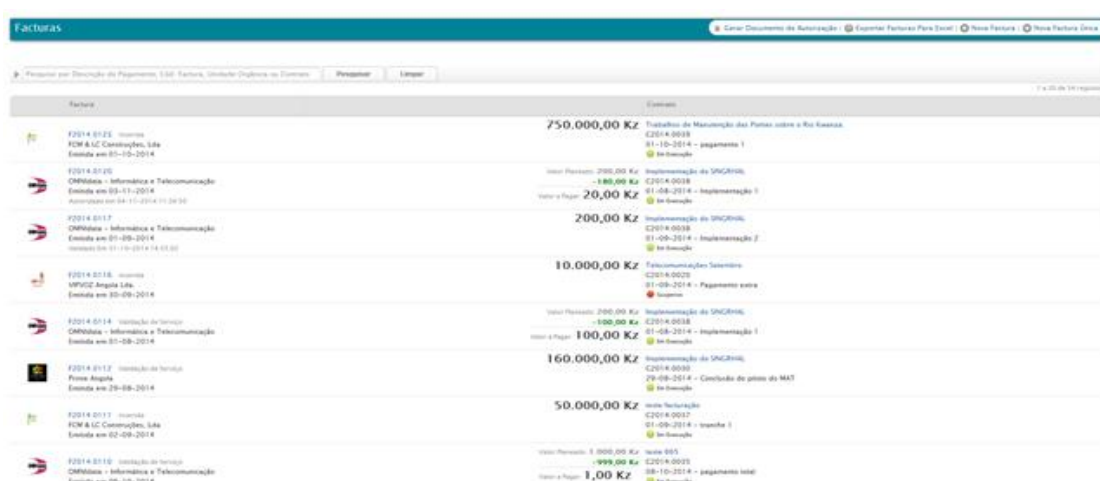
Figura 37 – Fluxo de vida de uma factura

## 4 – Problema para Aplicabilidade Prática

A gestão de facturas contempla as seguintes páginas principais:

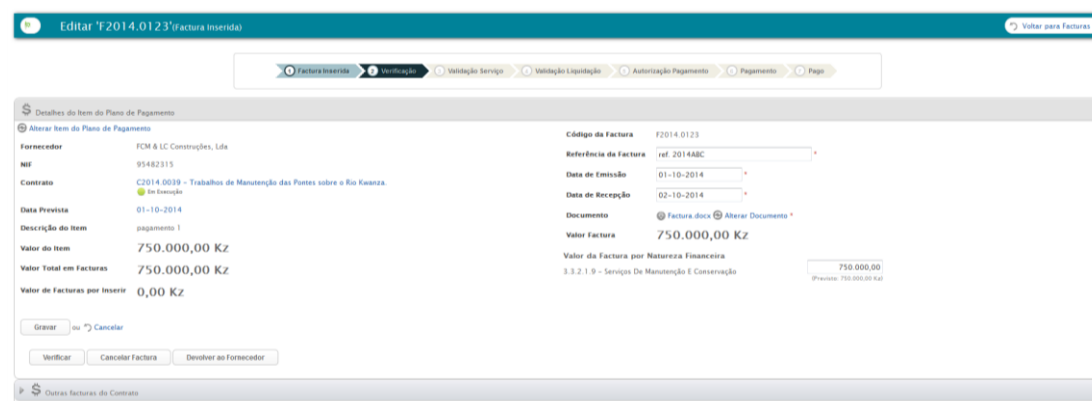
- **Listagem de facturas:** lista de todas as facturas existentes e activos no sistema. Nesta página é possível filtrar as facturas por diversos campos e ainda exportar a listagem para excel (de acordo com os filtros aplicados);
- **Páginas de edição de facturas:** páginas que permitem editar as informações das facturas.

As Figura 38 e Figura 39 mostram a listagem das facturas e uma página de edição das facturas.



Factura	Valor	Descrição
F2014.0123 - Inveniente FCM & LC Construções, Lda Emissão em 01-10-2014	750.000,00 Kz	Trabalhos de Manutenção dos Puentes sobre o Rio Kwanza. C2014.0039 01-10-2014 - pagamento 1
F2014.0125 - Inveniente OMNÍBUS - Informática e Telecomunicação Emissão em 04-11-2014 11:20:30	200,00 Kz Valor a Pagar: 200,00 Kz	Implementação do SNGR/MS. C2014.0038 01-08-2014 - Implementação 1
F2014.0117 - Inveniente OMNÍBUS - Informática e Telecomunicação Emissão em 01-09-2014	200,00 Kz	Implementação do SNGR/MS. C2014.0038 01-09-2014 - Implementação 2
F2014.0118 - Inveniente WVVIC Angola Lda Emissão em 20-09-2014	10.000,00 Kz	Telecomunicações Sotemore C2014.0039 01-09-2014 - Pagamento extra
F2014.0114 - Inveniente de Serviço OMNÍBUS - Informática e Telecomunicação Emissão em 01-09-2014	100,00 Kz	Implementação do SNGR/MS. C2014.0038 01-09-2014 - Implementação 1
F2014.0117 - Inveniente de Serviço Fomes Angola Emissão em 29-08-2014	160.000,00 Kz	Implementação do SNGR/MS. C2014.0039 29-08-2014 - Cancelado de parte do MAT
F2014.0117 - Inveniente FCM & LC Construções, Lda Emissão em 02-09-2014	50.000,00 Kz	Instalação de Serviços C2014.0037 01-09-2014 - parcela 1
F2014.0110 - Inveniente de Serviço OMNÍBUS - Informática e Telecomunicação Emissão em 09-10-2014	1,00 Kz	Instalação de Serviços C2014.0038 09-10-2014 - pagamento total

Figura 38 – Listagem de facturas



Editar 'F2014.0123' (factura inserida) Voltar para Facturas

Factura inserida | Verificação | Validação Serviço | Validação Liquidação | Autorização Pagamento | Pagamento | Pago

Detalhes do Item do Plano de Pagamento

Alterar Item do Plano de Pagamento

Forneecedor	FCM & LC Construções, Lda	Código da Factura	F2014.0123
NIF	95482315	Referência da Factura	ref. 2014ABC
Contrato	C2014.0039 - Trabalhos de Manutenção dos Puentes sobre o Rio Kwanza	Data de Emissão	01-10-2014
Data Prevista	01-10-2014	Data de Recepção	02-10-2014
Descrição do Item	pagamento 1	Documento	Factura.docx   Alterar Documento
Valor do Item	750.000,00 Kz	Valor Factura	750.000,00 Kz
Valor Total em Facturas	750.000,00 Kz	Valor da Factura por Natureza Financeira	750.000,00
Valor de Facturas por Inserir	0,00 Kz	3.3.2.1.9 - Serviços De Manutenção E Conservação	(Previsto: 750.000,00 Kz)

Generar ou Cancelar

Verificar | Cancelar Factura | Devolver ao Forneecedor

Outras Facturas do Contrato

Figura 39 – Página de edição de factura

Um dos aspectos importantes que este módulo permite é a associação de facturas a contratos. Embora não seja obrigatório, na maioria das vezes uma factura está associada a um contrato. A Figura 40 mostra como se pode associar uma factura a um contrato (que neste caso já tem uma factura associada a uma das suas actividades).

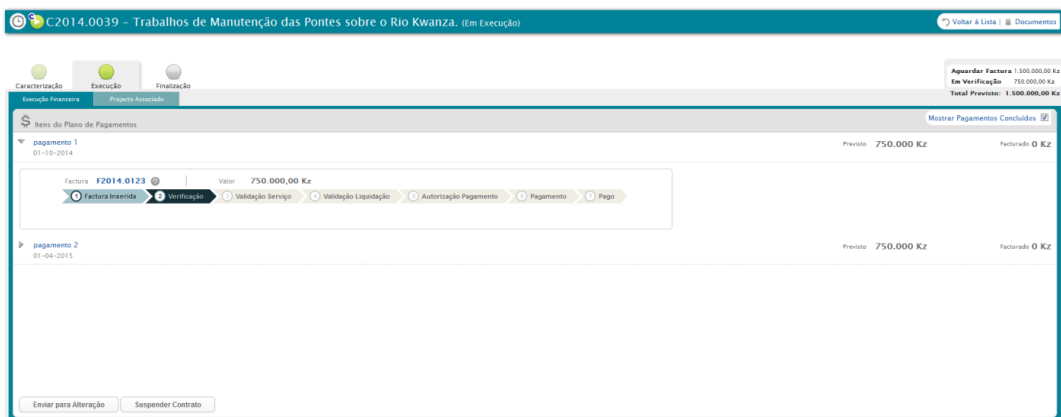


Figura 40 – Contrato com factura associada

Existem outros aspectos que poderiam sido abordados, no entanto, o exposto anteriormente é o suficiente para o leitor perceber o que o módulo de contratos faz, como e qual é o objectivo do mesmo.

#### 4.1.2 Sistema de Informação para Controlo e Acompanhamento de Projectos

O módulo projectos é constituído por um conjunto de funcionalidades que permitem assegurar a gestão detalhada de projectos ao longo do seu ciclo de vida. Encontra-se organizado por projectos vigentes e listagens operacionais de projectos vigentes, histórico de projectos, fichas de projecto e execuções financeira, física e de indicadores.

Os projectos classificam-se como vigentes ou arquivados de acordo com o estado em que encontram no decorrer do seu ciclo de vida. Consideram-se projectos vigentes os que se encontram em caracterização, validação, à espera de execução, execução, alteração ou em conclusão. Os projectos que se encontram nos estados cancelado, suspenso ou concluído consideram-se arquivados. A descrição de cada estado, tal como nos contratos, não é necessária, interessa saber que um projecto é vigente ou arquivado, dependendo do estado actual em que se encontra. A Figura 41 mostra o *flow* da transição de estados que um projecto pode sofrer ao longo do seu ciclo de vida.

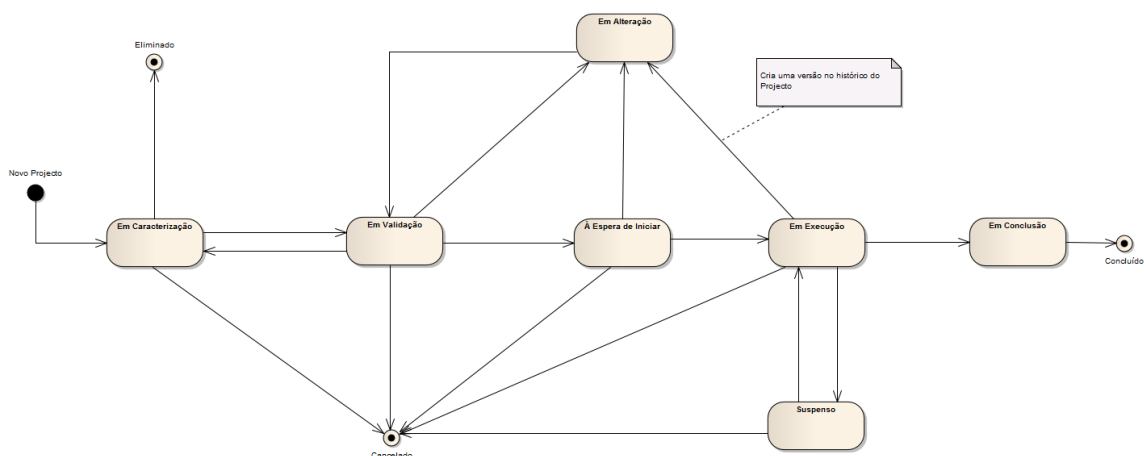


Figura 41 – Fluxo de vida de um projecto

A componente para a gestão de projectos disponibiliza aos utilizadores as seguintes páginas principais:

## 4 – Problema para Aplicabilidade Prática

- **Listagem de projectos:** lista de todos os projectos (vigentes ou arquivados) existentes e activos no sistema. Nesta página é possível filtrar os projectos por diversos campos e ainda exportar a listagem para excel (de acordo com os filtros aplicados);
- **Páginas de edição de projectos:** páginas que permitem editar as informações dos projectos. A informação a inserir está agrupada em vários separadores e em vários tabuladores;
- **Ficha de projecto:** página que permite visualizar todos os indicadores resumidos do projecto, sendo possível a sua exportação para pdf.

Por motivos de confidencialidade a página de listagem de projectos e a ficha de projecto não serão mostradas, no entanto as Figura 42, Figura 43 e Figura 44 mostram algumas páginas de edição de projecto.

Figura 42 – Página de edição de projecto

Figura 43 - Página de edição de projecto (2)

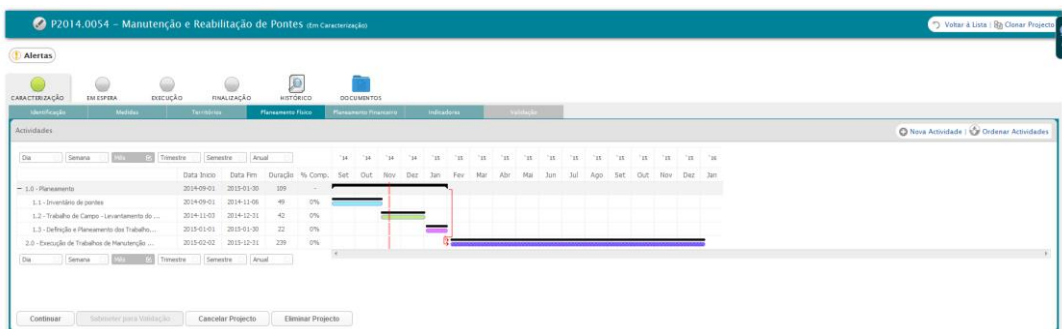


Figura 44 - Página de edição de projecto (3)

O módulo de gestão de projectos é o módulo mais importante da aplicação. Neste, o cliente gere todos os projectos que tem em carteira. Entre as várias informações associadas aos projectos, destaca-se o facto de os utilizadores poderem especificar os locais onde o projecto vai decorrer (Figura 43), gestão do planeamento físico do projecto (Figura 44) e associação de contratos a actividades do projecto no seu planeamento financeiro. Este módulo permite ainda fazer outras operações ao nível da gestão de projectos. No entanto, as funcionalidades apresentadas anteriormente são a que merecem maior destaque.

#### 4.1.3 Módulo de Gestão de BackOffice

O módulo de gestão de *backoffice* é responsável pela parametrização de várias entidades que depois podem ser associadas a projectos, contratos, facturas, etc. Por exemplo, um projecto pode estar incluído em  $n$  programas, os programas são geridos no *backoffice* (ora por inserção/edição através de formulários ou através da importação por um ficheiro de excel) e depois ficam disponíveis para associar aos projectos. O exemplo dado anteriormente aplica-se a várias entidades. A Figura 45 mostra uma página onde pode ser editada uma entidade (neste caso os fornecedores, que depois podem ser associados a contratos).

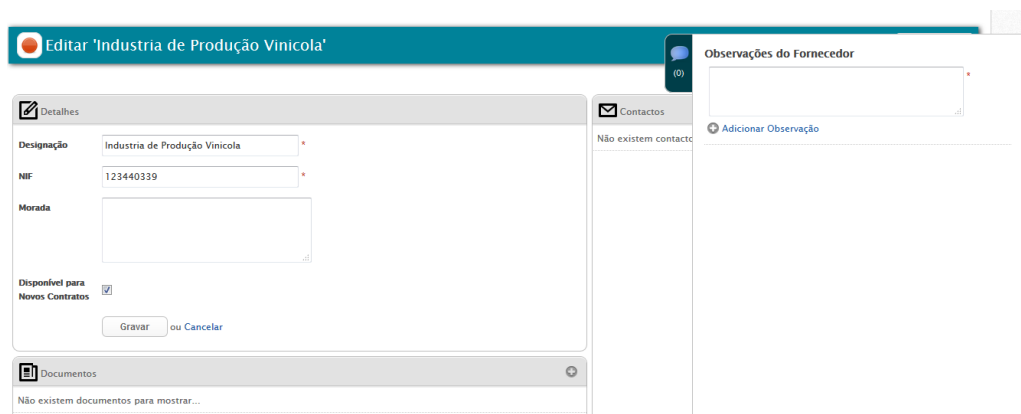


Figura 45 – Página de edição de uma entidade de negócio

O importante a reter sobre este módulo é que todos os indicadores, medidas e entidades de negócio associadas aos projectos, contratos e facturas são geridos neste. Estas entidades depois serão alvo de análise estratégica no módulo de *business intelligence*.

#### 4.1.4 Módulo de Business Intelligence – Portal do Conhecimento

Este módulo é responsável pela análise de BI feita aos dados sobre os projectos e contratos do sistema transaccional (composto pelos três módulos anteriores). Este portal de BI é conhecido como o portal do conhecimento.

O portal do conhecimento disponibiliza um conjunto de *dashboards* executivos que integram a informação de projectos e contratos, permitindo aferir o estado (quer global, quer de pormenor) dos projectos e contratos. A informação apresentada é respeitante ao dia anterior à análise, uma vez que, por razões de *performance*, o repositório de informação é actualizado durante a noite.

Relativamente aos projectos, estão disponíveis os seguintes *dashboards*:

- **Estratégico global por programa:** *dashboard* com o objectivo de representar a execução global dos projectos e identificar os programas mais desviantes. Possibilita também analisar a execução global dos projectos de um programa específico e identificar os subprogramas mais desviantes;
- **Estratégico global por medida:** *dashboard* com o objectivo de representar a execução global dos projectos e identificar as medidas mais desviantes. Possibilita também analisar a execução global dos projectos de uma medida específica e identificar os projectos mais desviantes;
- **Estratégico anual de projectos:** *dashboard* com o resumo da execução dos projectos, numa visão anual, e que permita identificar os projectos mais desviantes. Permite obter o resumo da execução por tipo de projecto;
- **Resumo de um projecto:** *dashboard* com um resumo do estado de um projecto.

Relativamente aos contratos estão disponíveis os seguintes *dashboards*:

- **Controlo de contratos:** *dashboard* que representa a capacidade de execução dos contratos firmados, em relação ao planeado;
- **Resumo de um contrato:** *dashboard* com um resumo do estado de um contrato.

Por questões de confidencialidade não é possível apresentar todos os conteúdos que formam este portal de BI. Seguidamente serão mostrados alguns componentes que formam os *dashboards* da *homepage* e da análise estratégica global por programa.

##### Homepage

A página principal do portal tem como objectivo dar uma visão geral sobre os projectos e indicadores dos mesmos. Um dos componentes do *dashboard* é a distribuição do número de projectos por estado. A Figura 46 mostra o referido componente.

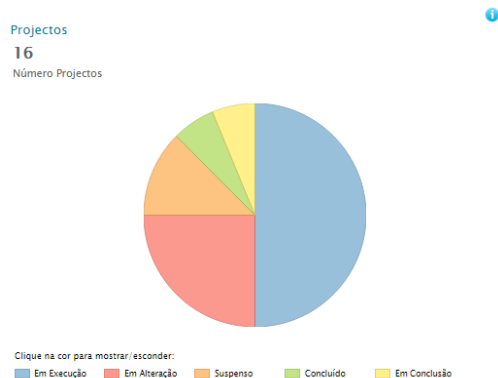


Figura 46 – Distribuição do número de projectos por estado

Através do componente anterior os utilizadores conseguem ter uma visão geral sobre os projectos e os seus estados.

Outro componente que é usado nesta *dashboard* é a análise financeira plurianual, que tem como objectivo mostrar o total orçamentado, contratado e facturado ao longo dos diversos anos. O desvio entre o total orçamentado, contratado e facturado é outra análise importante a fazer de modo a que a organização tenha consciência do cumprimento (ou incumprimento) dos orçamentos iniciais. A Figura 47 mostra este componente.

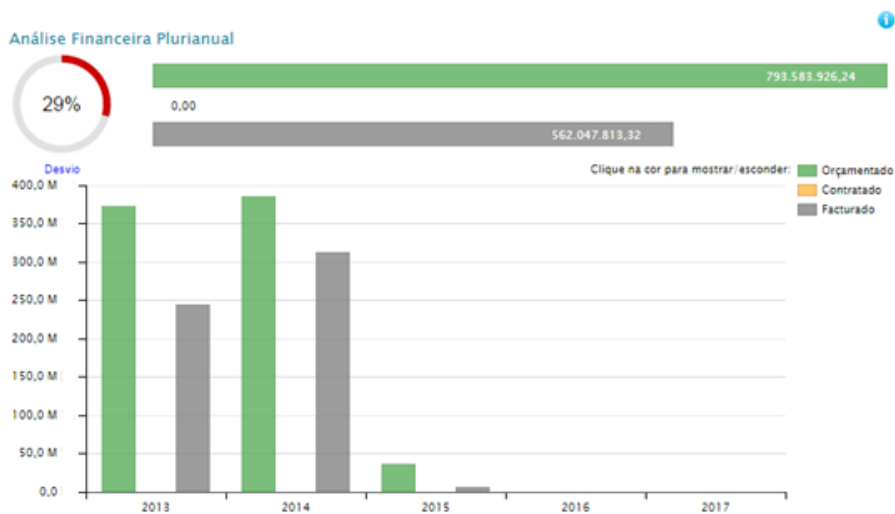


Figura 47 – Análise financeira plurianual

Na imagem anterior conseguimos ver, por ano, o total orçamentado, contratado (neste caso não existiu) e facturado de todos os projectos em carteira (gráfico de barras verticais no fundo da imagem). Conseguimos visualizar o somatório do orçamentado e facturado (barras horizontais no topo da imagem) e o desvio total existente (gráfico de progresso no canto superior esquerdo).

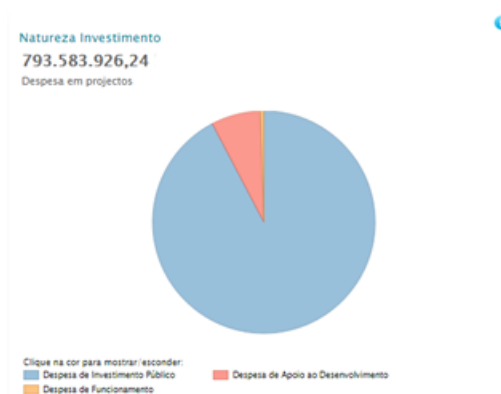
A distribuição dos projectos, por localização geográfica, está presente noutra componente, que é mostrado na Figura 48.



**Figura 48 – Distribuição geográfica dos projectos**

O componente anterior ajuda os utilizadores a perceberem a distribuição no espaço dos seus projectos em carteira. Através deste, conseguem perceber, por exemplo, as zonas onde se estão a desenvolver mais projectos (litoral vs interior, norte vs centro vs sul, etc.).

Um projecto tem associadas naturezas de investimento, o valor orçamentado de um projecto é então distribuído pelas mesmas, sendo assim importante perceber a distribuição financeira por natureza de investimento. O componente visualizado na Figura 49 mostra essa distribuição.



**Figura 49 – Investimento feito por natureza**

Existem outros componentes nesta *dashboard* que não podem ser divulgados. No entanto, os componentes apresentados ilustram o tipo de análises que são feitas na página principal. O objectivo é permitir aos utilizadores uma visão geral sobre os projectos que têm em carteira.

### **Análise Estratégica Global por Programa**

Trata-se de uma *Dashboard* que apresenta a execução global de todos os projectos aprovados e activos, cuja dimensão territorial englobe o território e programa seleccionados nos parâmetros de pesquisa. A *dashboard* apresenta dados de acordo com os parâmetros/filtros escolhidos, utilizando o form mostrado na Figura 50.

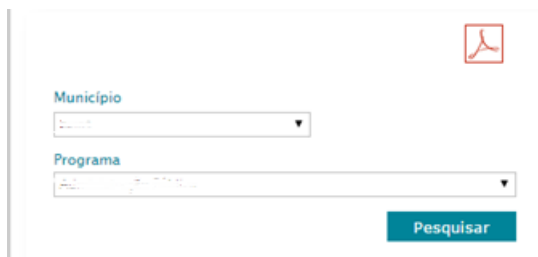


Figura 50 – Parâmetros/filtros de pesquisa do dashboard

Um dos componentes desta *dashboard* é a comparação entre os valores planeados e os valores executados financeiros de todos os projectos em carteira. Este componente também mostra o desvio entre o programado e o executado, sendo importante para os utilizadores acompanharem as liquidações dos valores programados. A Figura 51 mostra este componente.

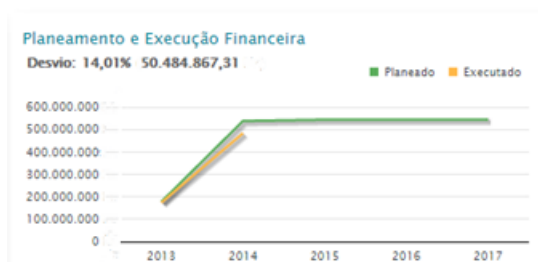


Figura 51 – Comparação entre o valor planeado e executado (com o desvio entre os valores)

Na figura anterior está representada a evolução dos valores planeados e executados ao longo dos anos (gráfico de linhas), sendo também mostrada a percentagem e valor do desvio que existe entre estes valores (por baixo do título do documento).

Outro componente presente nesta *dashboard* é a execução do investimento por programa. A Figura 52 mostra este componente.

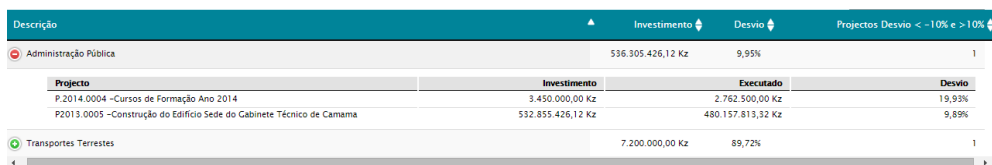


Figura 52 – Execução de investimento por programa

Na figura anterior consegue-se perceber que existem dois gráficos. O primeiro mostra o desvio acumulado até ao momento de cada programa (gráfico mais à esquerda); já o segundo mostra a execução financeira, ao longo dos anos, de cada programa. O componente ainda apresenta informação relativamente ao número de projectos em execução, percentagem de projectos por iniciar e percentagem de projectos por concluir (no topo do segundo gráfico).

O último componente deste *dashboard* é uma tabela que mostra algum detalhe sobre os projectos, agrupados por programa. Cada projecto, de cada programa, tem o investimento, o executado e o desvio entre os dois primeiros valores. O componente está representado na Figura 53.

## 4 – Problema para Aplicabilidade Prática



Descrição	Investimento	Desvio	Projectos Desvio < -10% e >10%
Administração Pública	536.305.426,12 Kz	9,95%	1
<b>Projecto</b>	<b>Investimento</b>	<b>Executado</b>	<b>Desvio</b>
P.2014.0004 – Cursos de Formação Ano 2014	3.450.000,00 Kz	2.762.500,00 Kz	19,93%
P2013.0005 – Construção do Edifício Sede do Gabinete Técnico de Camama	532.855.426,12 Kz	480.157.813,32 Kz	9,89%
Transportes Terrestes	7.200.000,00 Kz	89,72%	1

**Figura 53 – Detalhe sobre o investimento, execução e desvios dos projectos por programa**

Neste capítulo foram apresentados os diversos módulos que compõem a aplicação existente. Este enquadramento é importante para os leitores perceberem qual o sistema transaccional na qual vai ser aplicada a ferramenta de BI escolhida (3.8), assim como o que já existe na área de estudo (portal do conhecimento). Os capítulos seguintes apresentarão os desenvolvimentos efectuados no contexto deste trabalho.



## 5. Criação do Data Warehouse e dos Processos de ETL

A prova de conceitos prática que será feita na secção seguinte, com a criação de vários componentes, com a ferramenta SpagoBI, beberá dados de um *data warehouse* (ver Anexo 2 para consultar os fundamentos teóricos sobre *data warehousing*). O processo de criação do DW, os seus factos, dimensões e outros aspectos relacionados com a criação deste serão demonstrados nesta secção. O objectivo deste capítulo é os leitores perceberem algumas tabelas relacionais do sistema transaccional fonte e como vai ser criado o DW. Como aconteceu no capítulo anterior, existem questões de confidencialidade e de direitos de autor que terão que ser respeitadas, portanto serão apresentadas apenas as tabelas (e campos das mesmas) suficientes para os leitores perceberem a natureza dos dados trabalhados.

### 5.1 Sistema Transaccional

O sistema transaccional guarda todos os dados operacionais com uma regularidade diária. Todos os projectos, contratos e facturas são geridos neste sistema, posteriormente existem processos de ETL que extraem, transformam e carregam os dados da base de dados transaccional no DW que alimenta as consultas analíticas no Portal do Conhecimento existente. O DW que já existe, e que contém os dados que alimentam o Portal do Conhecimento, não deve ser modificado. Por isso, no âmbito deste trabalho e com a orientação da equipa de BI da organização Pessoas e Processos, criou-se um DW que alimentará os componentes que foram criados e que compõem a prova de conceitos prática.

A parte do sistema que foi utilizada para fazer a prova de conceitos prática foi a relativa aos projectos (os dados relativos aos processos contratos e às facturas não foram utilizados, por opção, devido ao facto dos projectos serem mais relevantes e esclarecedores para as análises). É então importante explicar algumas tabelas (do sistema transaccional fonte) que guardam informação relativa aos projectos.

A Figura 54 mostra algumas tabelas (e respectivos campos) que existem na base de dados transaccional. A tabela *projecto* guarda os dados base dos projectos, isto é, as suas características fundamentais como por exemplo: *código*, *designação*, *descrição*, *data início*, *data de fim*, etc. As tabelas *natureza*, *tipo* (projecto), *programa* e *subprograma*, *estado* (projecto) e *um\_unidade\_organica* (departamento/partição responsável pelo projecto) guardam dados de caracterização do projecto, isto é, o programa a que o projecto está inserido, o tipo de projecto, o estado actual do projecto, a natureza de investimento e o departamento a que pertence.

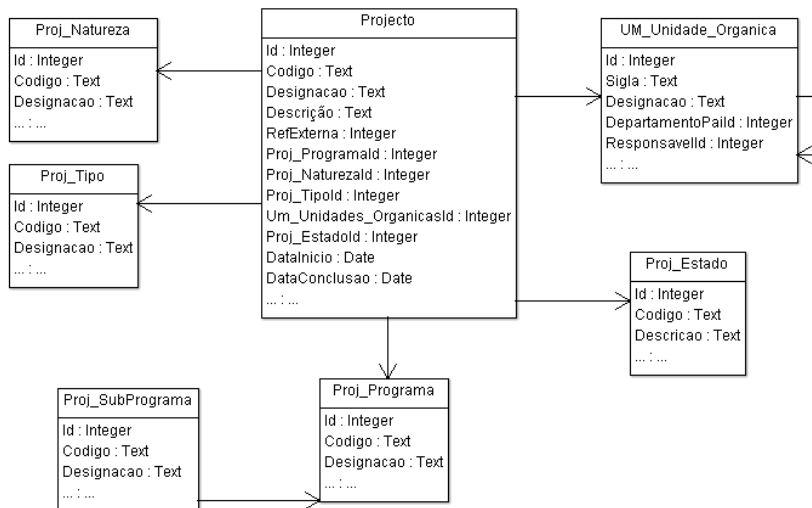


Figura 54 – Tabelas do sistema transaccional sobre os projectos

A Figura 55 mostra tabelas que, mais uma vez, adicionam dados de caracterização aos projectos. Um projecto pode estar associado a vários locais e ter várias medidas (tipo de política de quem investe). As tabelas *proj\_proj\_medida* e *proj\_proj\_territorio* fazem a ligação entre o projecto e as medidas (*proj\_medida*) e os territórios (*proj\_territorio*).

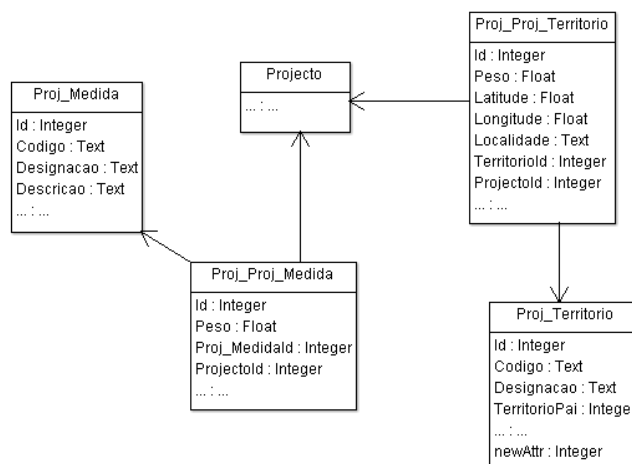


Figura 55 – Tabelas do sistema transaccional sobre os projectos (2)

A Figura 56 mostra as tabelas relacionadas com as actividades e naturezas dos projectos. A tabela *proj\_actividade* guarda, essencialmente, o nome das actividades, a tabela *proj\_projecto\_actividade* faz o mapeamento entre a tabela *projecto* e a *actividade*. Na tabela



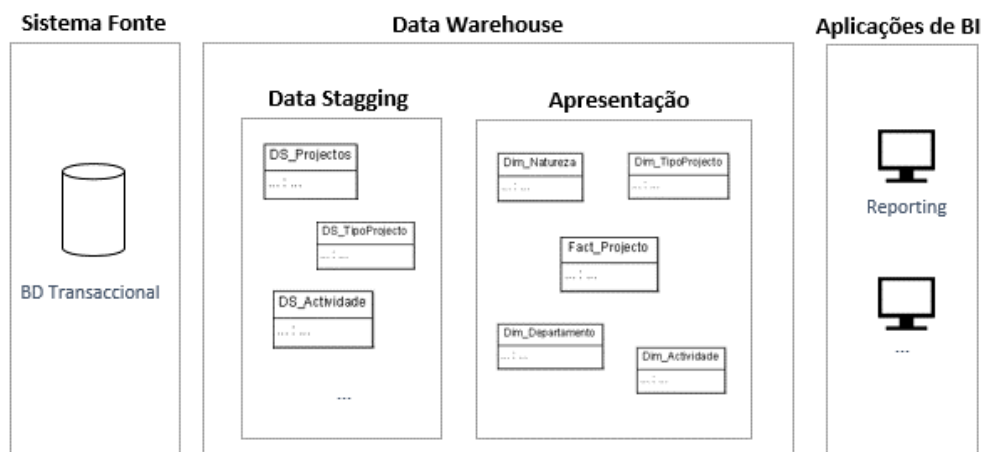


Figura 57 – Estrutura do DW desenvolvido

A construção do DW foi feita, sequencialmente, através da resposta a 4 perguntas essenciais, são elas [26][27]:

- Qual o processo de negócio a tratar?
- Qual a granularidade mais fina (nível de detalhe) dos dados necessária?
- Quais as dimensões necessárias?
- Quais os factos necessários?

A resposta à primeira pergunta, já foi dada anteriormente. O processo de negócio a tratar, neste trabalho, é o acompanhamento da evolução dos projectos ao longo do tempo. O acompanhamento da evolução dos projectos consiste, na aplicação da BI a desenvolver, na visualização de informação gráfica e de *reports* que respondam a algumas das perguntas seguintes:

- Qual a evolução, no que toca ao número, valores planeados e executados dos projectos ao longo dos anos?
- Como é a distribuição dos projectos, no que toca aos seus tipos e naturezas?
- Como é a distribuição geográfica dos projectos?

A resposta à segunda pergunta é muito importante para a concepção de qualquer DW. Deve-se analisar então com muito cuidado o nível de detalhe necessário do sistema a desenvolver. Junto da equipa de BI da organização, conseguiu-se identificar a granularidade mais fina do sistema. O DW guardará “fotografias” diárias dos projectos (e informação associada), havendo então uma periodicidade diária na extracção de informação do sistema fonte para o DW, sendo o nível de detalhe a informação relativa aos projectos.

A terceira questão foi respondida também com a ajuda da equipa de BI da organização. Tendo também em conta as dimensões já existentes no DW actual que alimenta o Portal do Conhecimento, e de acordo com a informação a mostrar na nova aplicação, identificaram-se, para o novo DW, as dimensões, que podem ser visualizadas na Figura 58.

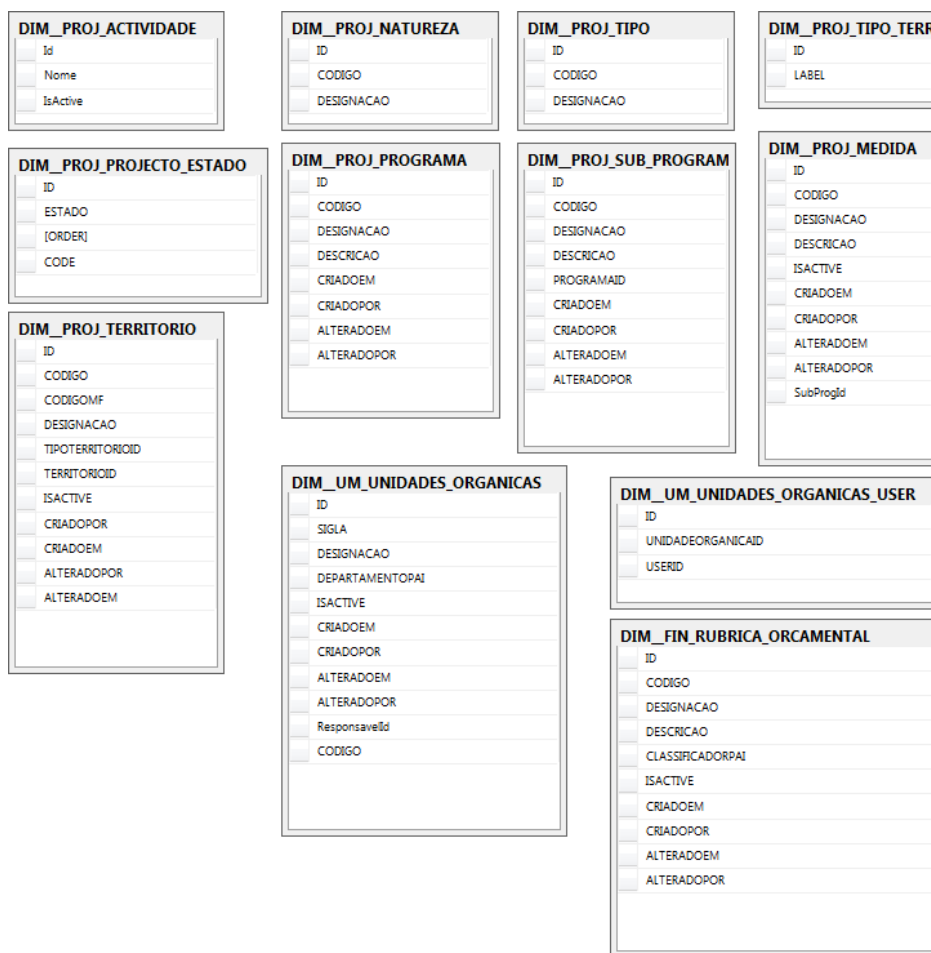


Figura 58 – Dimensões do DW desenvolvido

Na figura anterior estão todas as dimensões necessárias para as consultas que serão feitas na aplicação de BI desenvolvida. Estas dimensões, como já foi referido, foram identificadas de acordo com as dimensões já existentes no DW existente. No entanto, as dimensões (e atributos de cada) que foram criadas e estão representadas na figura anterior, não são cópias das já existentes. Existem atributos, que para as análises a fazer na nova aplicação não são necessárias, sendo então eliminadas. Existem outras dimensões no DW já existente, que não foram replicadas no novo DW, pelo mesmo motivo, ou seja, não se justificava a sua inclusão.

A quarta questão, mais uma vez, foi respondida com a ajuda da equipa de BI da organização. A identificação dos factos a analisar é um passo crucial no desenvolvimento de um DW. Como o próprio nome indica, as tabelas de factos, registam os factos do negócio a analisar. Já se identificou, anteriormente, que o processo de negócio a tratar é o acompanhamento e evolução dos projectos. Consegue-se então identificar, de forma clara, que os projectos são um dos factos, talvez o principal, a analisar, mas existem outros. Na Figura 59 podem ser visualizadas as tabelas de factos identificadas e que foram criadas no novo DW.

FACT_PROJECTO_TERRITORIOS	FACT_PROJECTO_MEDIDA	FACT_PROJECTO_ACT_FISIC_EXEC	FACT_PROJECTO
AUX_ETLPROCESSID	AUX_ETLPROCESSID	AUX_ETLPROCESSID	AUX_ETLPROCESSID
PROJECTO_ID	PROJECTO_ID	PROJECTO_ID	PROJECTO_ID
ANO	ANO	ANO	ANO
NOME	NOME	NOME	NOME
DESCRICAO	DESCRICAO	DESCRICAO	DESCRICAO
DATAINICIO	DATAINICIO	DATAINICIO	DATAINICIO
DATAFIM	DATAFIM	DATAFIM	DATAFIM
CRIADOPOR	CRIADOPOR	CRIADOPOR	CRIADOPOR
CRIADOEM	CRIADOEM	CRIADOEM	CRIADOEM
ALTERADOPOR	ALTERADOPOR	ALTERADOPOR	ALTERADOPOR
ALTERADOEM	ALTERADOEM	ALTERADOEM	ALTERADOEM
ISACTIVE	ISACTIVE	ISACTIVE	ISACTIVE
ISINVESTIMENTOPUBLICO	ISINVESTIMENTOPUBLICO	ISINVESTIMENTOPUBLICO	ISINVESTIMENTOPUBLICO

FACT_PROJECTO_ACT_FISIC	FACT_PROJECTO_ACT_FINANC	FACT_PROJECTO_ACT_FINANC_EXEC
AUX_ETLPROCESSID	AUX_ETLPROCESSID	AUX_ETLPROCESSID
PROJECTO_ID	PROJECTO_ID	PROJECTO_ID
ANO	ANO	ANO
NOME	NOME	NOME
DESCRICAO	DESCRICAO	DESCRICAO
DATAINICIO	DATAINICIO	DATAINICIO
DATAFIM	DATAFIM	DATAFIM
CRIADOPOR	CRIADOPOR	CRIADOPOR
CRIADOEM	CRIADOEM	CRIADOEM
ALTERADOPOR	ALTERADOPOR	ALTERADOPOR
ALTERADOEM	ALTERADOEM	ALTERADOEM
ISACTIVE	ISACTIVE	ISACTIVE
ISINVESTIMENTOPUBLICO	ISINVESTIMENTOPUBLICO	ISINVESTIMENTOPUBLICO

Figura 59 - Factos do DW desenvolvido

Mais uma vez, a equipa de BI da organização validou os factos identificados, as tabelas de factos baseiam-se nas já existentes no DW actual. No entanto, tal como aconteceu nas tabelas dimensão, nem todos os factos foram replicadas no novo DW desenvolvido, foram identificados apenas aqueles que são necessários para as análises a fazer na aplicação de BI. Na figura anterior não são perceptíveis todos os campos das tabelas de factos devido ao facto destas tabelas serem, no que toca ao número de campos, muito extensas.

A aplicação de BI deve ter a capacidade de mostrar o histórico de dados dos projectos. Todas as alterações que são feitas num projecto, ao longo do tempo, devem estar guardadas no DW desenvolvido. Juntamente com a equipa de BI, foi identificado o tipo das tabelas de factos que deve ser utilizado, este é conhecido como de *snapshot*. As tabelas de facto do tipo *snapshot*, guardam “fotografias” dos dados num determinado momento. Já foi anteriormente referido, que os dados devem ser carregados no DW diariamente, ou seja, as tabelas de factos vão guardar “fotografias” diárias dos projectos.

No início do desenvolvimento do DW foram apresentadas pela equipa de BI da organização, duas alternativas para o desenho do mesmo, de modo a guardar o histórico dos dados e as alterações que os mesmos podem sofrer ao longo do tempo. A primeira baseava-se na construção de tabelas dimensão que lidem com as dimensões em alteração lenta. Com esta abordagem, todas as alterações que são feitas aos dados das dimensões são guardadas nas dimensões. A segunda baseia-se na construção de tabelas de facto que guardam todas as alterações dos dados do projecto, incluindo as alterações aos dados das dimensões. Com esta abordagem, as dimensões apenas guardam os dados mais actuais do sistema transaccional, mas o histórico destas é, na mesma mantido nas tabelas de facto. Entre as duas abordagens sugeridas, optou-se pela a segunda abordagem. Esta foi a solução criada pela equipa de BI

da organização na construção do DW existente. Assim, qualquer dúvida que surgisse na construção do novo DW, podia ser mais rapidamente e facilmente clarificada pela equipa de BI.

A abordagem escolhida, no que toca ao processo de ETL (mais detalhes na secção seguinte), a fazer diariamente, é constituída pelos seguintes passos:

1. Eliminar todos os dados existentes nas dimensões no DW, e inserir os dados mais actuais do sistema transaccional;
2. Inserir os dados sobre os factos de negócio a analisar, numa primeira fase, na área de *data staging* (área temporária de dados);
3. Transformar os dados existentes na área de *data staging* e inserir os mesmos nas tabelas de factos;
4. Eliminar todos os dados das tabelas de *data staging*.

Os passos anteriores formam o fluxo de dados que deve existir entre o sistema transaccional e o DW. Este fluxo foi respeitado nos processos de ETL desenvolvidos, e que serão explanados na secção seguinte.

No que toca aos passos descritos, é importante esclarecer a importância da área de *data staging* [27]. Não é obrigatório que esta área exista. No entanto, a utilização desta, simplifica a transformação dos dados, e por este motivo, optou-se por utilizar esta área. Esta tem tabelas que são, fisicamente, muito idênticas às tabelas transaccionais. O objectivo destas é que os dados transaccionais estejam, temporariamente no DW, tal e qual como são no sistema fonte. Estes dados, existindo fisicamente no DW, são mais fáceis de transformar. Numa primeira fase, os dados das tabelas de *data staging* estão normalizados, isto faz com que o processo de extracção seja mais rápido. A desnormalização dos dados, e sua inserção nas tabelas de factos, é feita já no DW, numa segunda fase. Não existindo esta área, seria necessário que os dados que vêm do sistema fonte, já viessem preparados, ou seja, desnormalizados, para serem inseridos directamente nas tabelas de factos. Em alternativa, os processos de ETL podiam usar mecanismos que permitissem que os dados fossem, antes de serem carregados nas tabelas de factos, transformados temporariamente (utilizando, por exemplo, tabelas temporárias ou repositórios de dados em memória), mas o tempo de extracção (que incluía posteriormente a transformação e carregamento nas tabelas de facto) dos dados transaccionais para o DW demoraria muito mais tempo e requeria mais capacidade de processamento.

Depois de carregados os dados nas tabelas de facto, já não precisam nas tabelas de *data staging*, então estes são apagados, isto faz com que, antes de um processo de ETL ser executado (com periodicidade diária) as tabelas de *data staging* estejam vazias, diminuindo assim o tempo de inserção dos dados nestas.

Outro aspecto que é importante referir é que os dados recebidos do sistema transaccional são responsabilidade da equipa deste. Caso haja algum problema de coerência, integridade ou qualidade dos dados então a equipa de BI não é responsável pelo mesmo, o mesmo acontecendo também no âmbito deste trabalho, ou seja, problemas de qualidade dos dados, incosistências, integridade, polissemias, etc., não serão tratadas no processo de ETL criado. Neste projecto, a equipa do sistema transaccional, forneceu os dados através de *views*, criadas para o efeito, no sistema fonte. Os dados

que foram disponibilizados nas *views* foram os identificados e pedidos para usar no âmbito deste trabalho, com a supervisão e ajuda da equipa de BI da instituição.

### 5.3 Processos de ETL

A população do DW foi feita através da criação de vários processos de ETL. Todos os processos criados, juntos, criam um único processo responsável pela extracção, transformação e carregamento dos dados do sistema transaccional para o DW. A equipa de BI da instituição supervisionou a criação destes processos. Os processos criados, no que toca à lógica geral e ao fluxo seguido, são muito idênticos aqueles que foram criados para popular o DW existente e que alimenta, o também já existente, Portal do Conhecimento.

A equipa de BI utiliza uma ferramenta comercial para criar os processos de ETL, por motivos de confidencialidade não será apresentada a mesma. Na secção 3.8 foi dito que a ferramenta SpagoBI foi a escolhida para criar a aplicação de BI. Como foi apresentada na secção 3.6.5, o SpagoBI utiliza a ferramenta Talend Open Studio para tratar da integração dos dados. O TOS permite a criação de processos de ETL. Na secção 3.2.1 foram apresentadas as características desta ferramenta. Antes de criar os processos de ETL com o TOS foi preciso instalar esta ferramenta, em [http://www.talendforge.org/wiki/doku.php?id=doc:installation\\_guide](http://www.talendforge.org/wiki/doku.php?id=doc:installation_guide) podem-se ver todos os passos seguidos para a instalação da mesma.

#### 5.3.1 Selecção do Workspace e do Projecto

Como foi referido na secção 3.2.1, o TOS trabalha à base de *workspaces*. Um *workspace*, como já foi referido, é uma organização lógica de trabalho, sendo habitual cada projecto de ETL ter um. Assim, antes de se criarem os processos de ETL, foi criado um *Workspace*, aquele que já vem configurado por omissãodefeito. Depois de seleccionado o *Workspace*, foi criado o projecto. A criação deste pode ver-se na Figura 60 e na Figura 61.

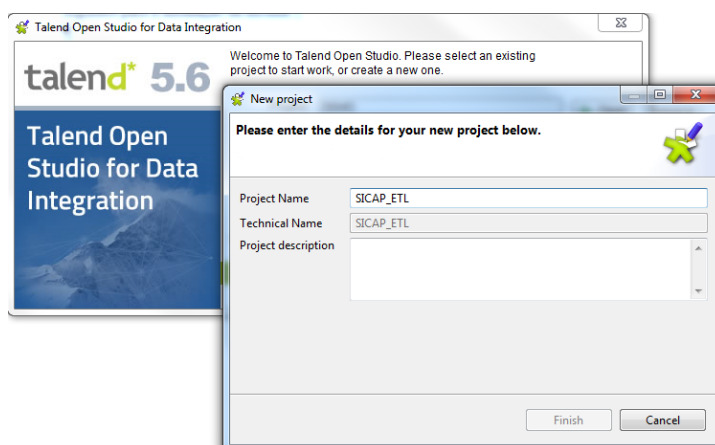


Figura 60 – Criação do projecto no workspace por omissão

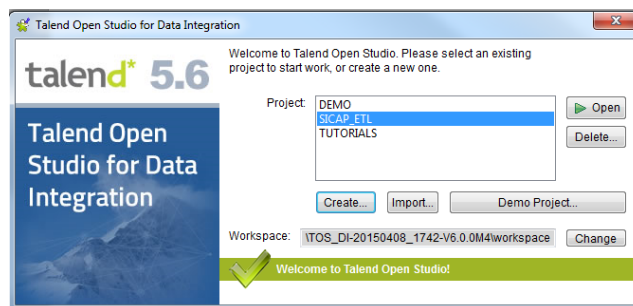


Figura 61 - Criação do projecto no workspace por omissão (2)

Nas figuras anteriores consegue-se perceber que foi criado o projecto com o nome *SICAP\_ETL*, sendo neste que foram criados todos os processos de ETL.

### 5.3.2 Criação dos Metadados e Ligação ao Sistema Fonte

O TOS permite a criação de metadados que ajudam a criação dos processos de ETL. É possível criar metadados para ligações a bases de dados, ficheiros excel, ficheiros xml, *webservices*, servidores de ficheiros, etc. Na Figura 62 estão visíveis todos os tipos de metadados possíveis de criar no TOS.

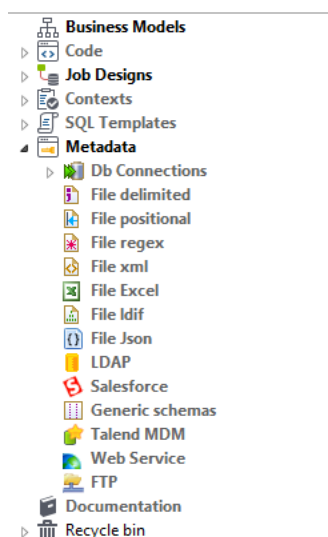


Figura 62 – Tipos de metadados possíveis de criar no TOS

Neste trabalho, apenas foram usados metadados para ligações a bases de dados. Com o TOS é possível dar a ligação a uma base de dados e escolher os objectos a mapear para o projecto. No projecto de ETL criado, foram escolhidas duas ligações a bases de dados. A primeira ligação é ao sistema transaccional. Com esta ligação foram mapeadas todas as tabelas e seus campos. A segunda ligação é ao DW. Tal como na primeira, todas as tabelas e seus campos foram mapeados para o sistema. Na Figura 63 e na Figura 64 podem-se ver as ligações feitas no TOS.

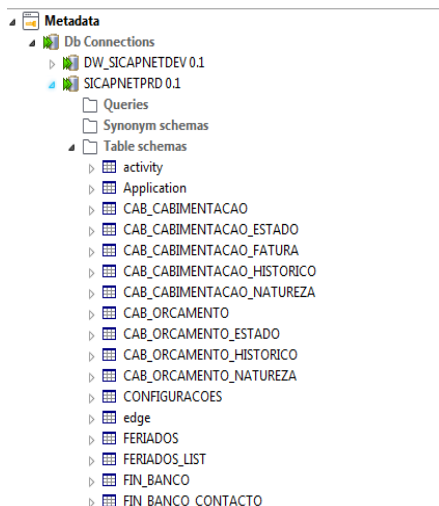


Figura 63 – Metadados da ligação à base de dados transaccional

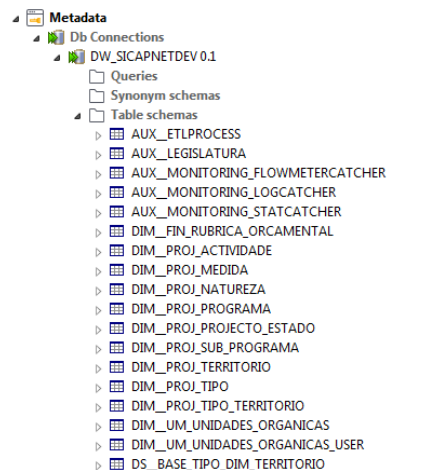


Figura 64 – Metadados da ligação ao DW

Nas figuras anteriores estão apenas visíveis algumas tabelas dos dois sistemas que foram mapeadas. Todas as tabelas e seus campos, das duas bases de dados, foram mapeados pelo TOS. Este mapeamento ajudou na criação dos processos de ETL, visto que, com este mapeamento, a extracção e carregamento de dados nas tabelas tornou-se menos moroso. As tabelas alvo de extracção e carregamento de dados, tanto da base de dados relacional como do DW, foram escolhidas através dos metadados no TOS. Assim, todos os campos das tabelas (os seus tipos, limites de tamanho, etc.) foram automaticamente. Com este processo automático, não foi necessário criar todos os campos nos componentes dos processos de ETL, pois o TOS, através dos metadados, apresentou-os no imediato.

### 5.3.3 Criação dos Processos

Na secção 5.2 foram apresentados os passos que devem seguidos, de acordo com a abordagem escolhida no que toca à arquitectura do DW, para inserir os dados no DW. A criação do processo de ETL que trata de toda a extracção, transformação e carregamento de dados do sistema transaccional para o DW, foi feita usando vários subprocessos. Cada subprocesso trata de uma parte lógica e operacional dos dados a trabalhar. Todos os processos juntos criam o processo geral para a integração dos dados. Na Figura 65 está representado o fluxo de dados que foi seguido em todo o processo de ETL.

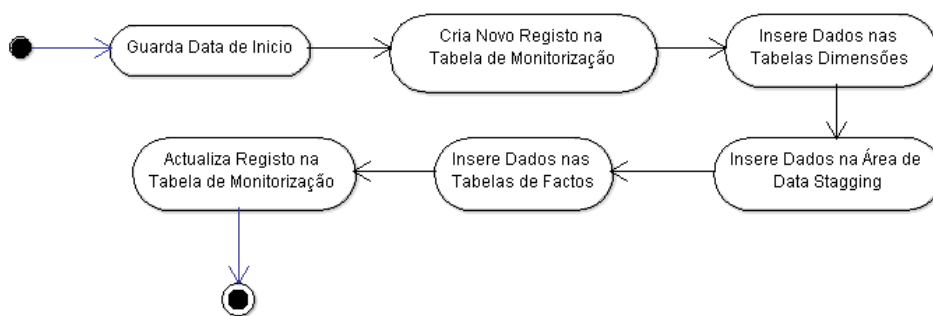


Figura 65 - Diagrama de actividades com o fluxo de ETL seguido

As duas primeiras actividades tratam da criação de um novo registo numa tabela (*AUX\_ETLPROCESS*), existente no DW, que auxilia e monitoriza os processos de ETL. Esta tabela tem como objectivo identificar cada processo de ETL, guardando ainda as datas e horas de início e fim de cada processo. Depois de inserido o registo na tabela de monitorização, os registos das dimensões são apagados e inseridos os dados mais actualizados. Este passo corresponde à terceira actividade do diagrama. A quarta actividade do diagrama, representa, a inserção dos dados do sistema transaccional na área de *data staging*. Depois de inseridos os dados na área temporária do DW, os mesmos são transformados e inseridos nas tabelas de factos. Este passo corresponde à quinta actividade. A sexta actividade corresponde à actualização do registo da tabela de monitorização. O registo criado no início do processo é actualizado com a data de fim do mesmo. O processo de ETL, logicamente, considera-se terminado nesta actividade. Por fim, apagam-se todos os dados da área de *data staging*. Estes dados já não são úteis e a sua eliminação faz com que as tabelas desta área estejam sempre vazias no início de um processo de ETL, optimizando assim o tempo de inserção de dados nestas, optimizando também o tempo de processamento de todo o processo de ETL.

Foram criados ao todo trinta e dois subprocessos de ETL. Em conjunto tratam de todo o processo de integração dos dados da base de dados transaccional no DW. Serão apresentados os subprocessos mais importantes para a compreensão do processo de integração de dados.

O primeiro subprocesso é aquele que é chamado pelo SpagoBI. É este subprocesso que desencadeia a primeira chamada a outro subprocesso, que em cadeia, através de chamadas entre outros subprocessos, chega à ao último subprocesso. Este será depois responsável pela finalização do processo geral de integração de dados. O primeiro subprocesso pode ser visto na Figura 66.

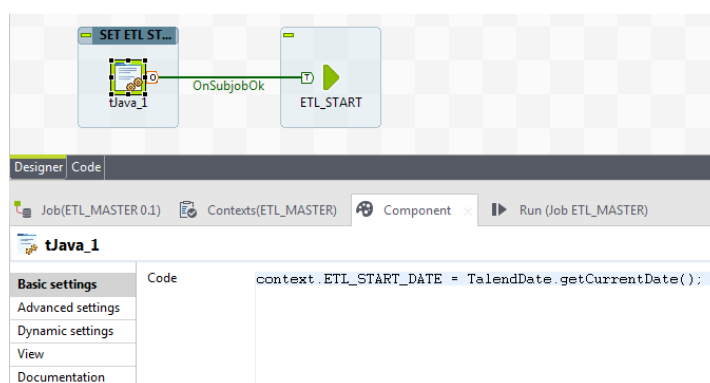


Figura 66 – Subprocesso inicial (ETL\_MASTER)

## 5 – Criação do Data Warehouse e dos Processos de ETL

O primeiro componente, trata de inserir, numa variável de contexto, a data de início do processo de integração de dados. Esta variável é depois passada para o segundo componente. Este último chama o subprocesso *ETL\_START*.

O processo *ETL\_START* é responsável pela criação do registo na tabela de monitorização dos processos de ETL, que existe fisicamente no DW. Este processo pode ser visto Figura 67.

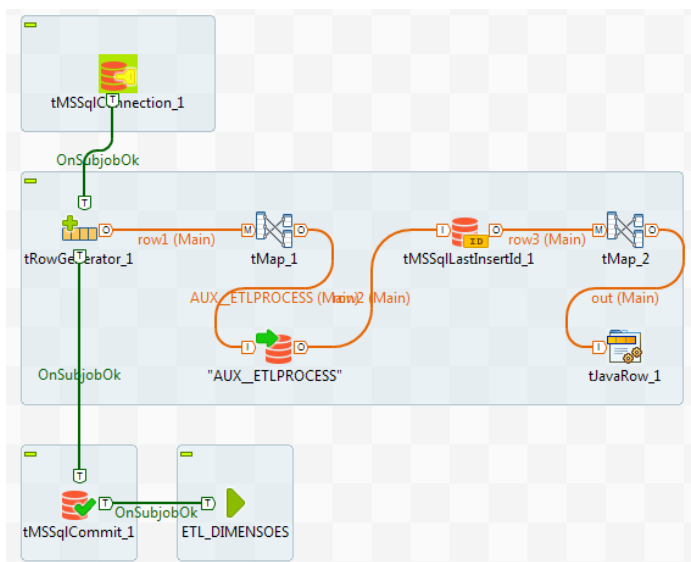


Figura 67 – Subprocesso ETL\_START

O primeiro componente, da figura anterior, faz uma ligação directa à base de dados. O segundo cria um novo registo em memória com a data actual e a data de início (inserido na variável de contexto do subprocesso *ETL\_Master*). O terceiro componente faz o mapeamento entre os dados da linha criada no segundo componente e dados da tabela *AUX\_ETLPROCESS*. Este tipo de componente, chamado de *tMap*, é muito utilizado no TOS. O *tMap* faz mapeamento entre dados de componentes diferentes. Neste caso, como se pode ver na Figura 68, fez-se o mapeamento dos dados criados no segundo componente (fonte origem) e dos dados da tabela *AUX\_ETLPROCESS* (fonte destino).

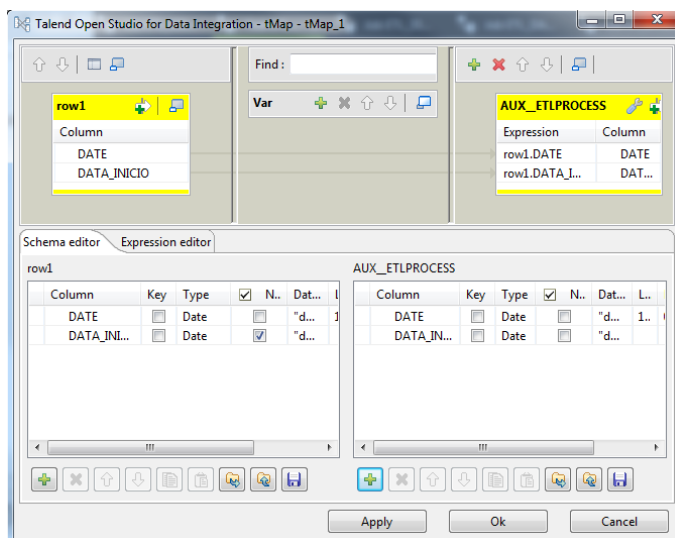


Figura 68 – Mapeamento de dados entre fonte origem e destino, usando o componente tMap

O quarto componente representa uma tabela existente no DW. Neste caso, a tabela de monitorização *AUX\_ETLPROCESS*. Este componente serve para o componente anterior ter uma fonte destino e para inserir um registo nesta tabela (com a data actual e a data de início do processo de ETL). O quinto componente, consegue obter o último *ID* existente numa tabela. Neste caso, como o componente de entrada é a tabela *AUX\_ETLPROCESS*, o componente retorna na sua saída o último *ID* desta tabela. O sexto componente faz novamente um mapeamento de dados. Neste caso em particular, é usado para o sétimo componente usar o último *ID* inserido na tabela *AUX\_ETLPROCESS* (registo inserido através do quarto componente). O sétimo componente guarda numa variável de contexto o *ID* do registo inserido na tabela de monitorização. Este *ID* será depois utilizado no subprocesso final, que depois actualizará o registo da tabela de monitorização, de modo a que seja inserida a data de finalização do processo de ETL. O oitavo componente faz o *commit* da transacção na base de dados. Este *commit* é necessário porque no início do subprocesso, o primeiro componente fez uma ligação directa à base dados. Este componente cria uma transacção, ou seja, todas as operações de inserção, actualização e eliminação de dados só são definitivas se existir um *commit* da transacção. Foi necessário fazer uma ligação directa à base de dados, de maneira a que se conseguisse obter, antes do fim do subprocesso, o *ID* do registo inserido na tabela de monitorização. O último componente chama o subprocesso *ETL\_DIMENSOES*.

O subprocesso *ETL\_DIMENSOES* é um agregador de subprocessos. Cada um dos subprocessos existente no *ETL\_DIMENSOES*, é responsável por tratar da inserção dos dados de cada uma das dimensões do DW. Este subprocesso pode ser visto na Figura 69.

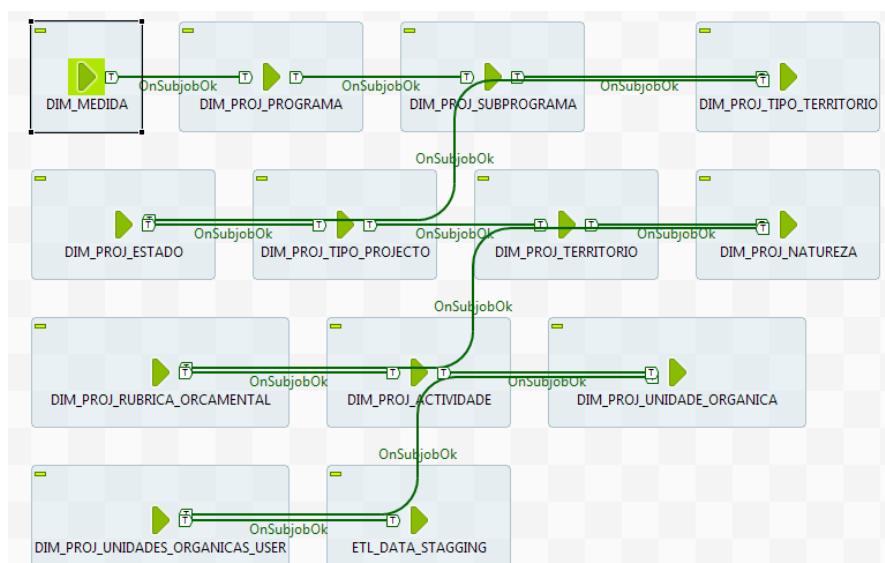


Figura 69 – Subprocesso *ETL\_DIMENSOES*

Cada componente do subprocesso *ETL\_DIMENSOES* é uma chamada a outro subprocesso. Os primeiros doze componentes chamam outros, que como já foi referido, tratam da inserção de dados nas tabelas dimensão correspondente. O último componente, chama o subprocesso responsável pela inserção de dados na tabelas da área de *data staging* do DW.

Todos os subprocessos que tratam da inserção dos dados nas dimensões são muito idênticos. Na Figura 70 pode ser visto um destes subprocessos.

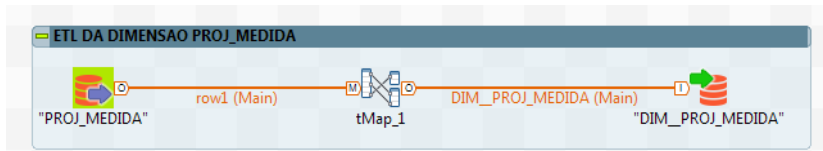


Figura 70 – Subprocesso que insere dados na dimensão DIM\_PROJ\_MEDIDA

O primeiro componente especifica a fonte origem. Neste caso, a tabela transaccional DIM\_PROJ\_MEDIDA. O segundo faz o mapeamento entre a fonte origem (componente anterior) e a fonte destino (terceiro componente). Este componente já foi mostrado anteriormente, tem o nome de tMap e é responsável por fazer mapeamento de dados. Na Figura 71 pode ser visto, para este caso particular, como foi feito o mapeamento dos dados entre a tabela transaccional e a dimensão.

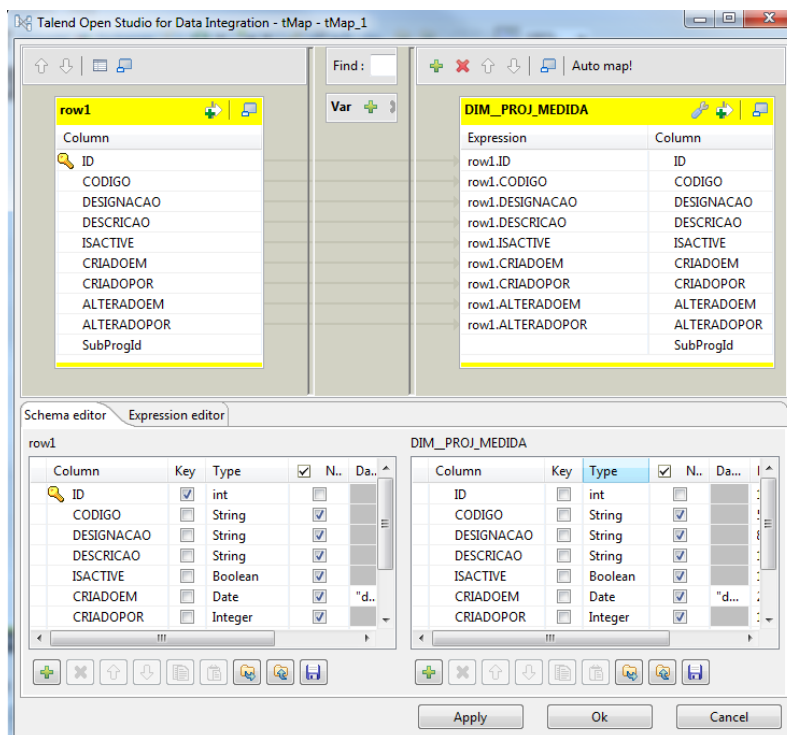


Figura 71 – Mapeamento entre a tabela transaccional e a dimensão do DW

O terceiro componente representa a dimensão já referida. Este componente foi configurado, de modo a que no momento da inserção dos dados, todos os dados existentes na dimensão, fossem apagados antes da inserção. Na Figura 72 consegue-se perceber que na opção “Action on table” escolheu-se fazer a eliminação dos dados já existentes (através do “Truncate table”). Depois dos dados eliminados, o componente trata de inserir os dados que vêm do segundo componente, este que por sua vez tem como fonte origem os dados da tabela transaccional escolhida, neste caso a tabela PROJ\_MEDIDA.

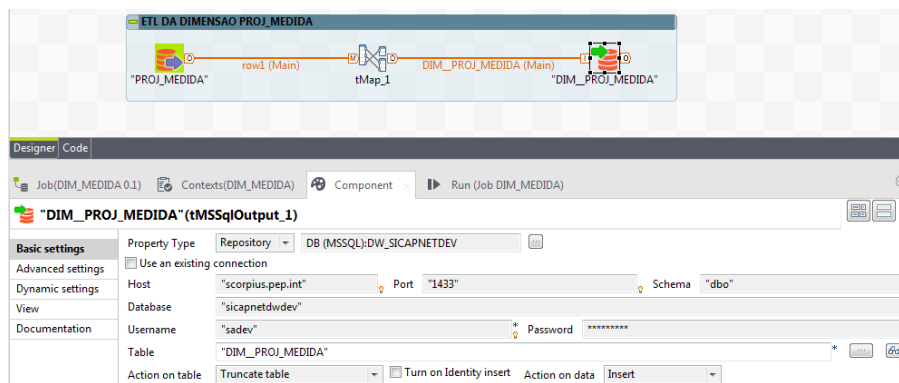


Figura 72 – Eliminação dos dados existentes numa tabela

Este subprocesso, ao contrário do *ETL\_START*, não faz nenhum *commit* ao DW, isto deve-se ao facto de não se ter feito, explicitamente, uma ligação directa à base de dados. Isto faz com que todas as operações feitas aos dados, durante um processo de ETL, sejam automaticamente definitivas.

Como já foi referido e pode ser visto na Figura 69, o subprocesso *ETL\_DIMENSOES*, no seu último componente chama o subprocesso *ETL\_DATA\_STAGGING*. Este é muito idêntico ao subprocesso *ETL\_DIMENSOES*, sendo também num agregador de chamadas a subprocessos que, individualmente, tratam da inserção de dados numa tabela de *data staging*. Os subprocessos que inserem dados nas tabelas de *data staging* são também muito parecidos ao subprocesso *DIM\_PROJ\_MEDIDA*. Nestes também é feita uma eliminação dos dados já existentes, pois como já foi referido os dados que existem nestas tabelas temporárias, antes da inserção de novos dados, já não são necessários pois já foram processados anteriormente. Consistem em três componentes, o primeiro para definir a origem dos dados, o segundo para fazer o mapeamento e o terceiro para definir o destino dos dados, tendo como base o mapeamento realizado. No fim do subprocesso *ETL\_DATA\_STAGGING*, é chamado o subprocesso *ETL\_FACTOS*. Este é idêntico ao subprocesso *ETL\_DATA\_STAGGING*, com a excepção da operação sobre a fonte destino pois os dados já existentes são mantidos e nunca eliminados. Devido ao facto dos subprocessos *ETL\_DATA\_STAGGING* e *ETL\_FACTOS* serem idênticos ao *ETL\_DIMENSOES*, tendo este já sido explicado, não serão então explanados os dois primeiros, nem os subprocessos que tratam da inserção dos dados nas tabelas de *data staging* e das tabelas de factos pois são também muito parecidos ao subprocesso *DIM\_PROJ\_MEDIDA*. No fim do subprocesso *ETL\_FACTOS* é chamado o último subprocesso.

O último subprocesso tem o nome de *ETL\_END*. Este subprocesso tem como objectivo actualizar o registo inserido na tabela de monitorização, de modo a que este fique com a data de conclusão do processo. Na Figura 73 pode ser visto este subprocesso.

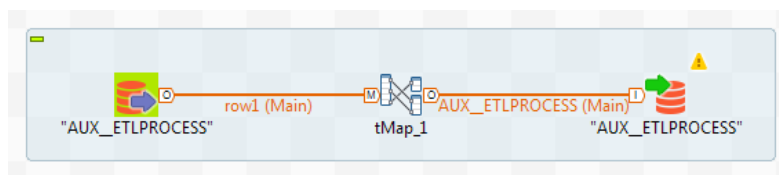


Figura 73 – Subprocesso ETL\_END

O primeiro componente especifica a tabela fonte, neste caso a *AUX\_ETLPROCESS*. O segundo faz o mapeamento entre a fonte origem (componente anterior) e a fonte destino (terceiro componente). O componente que faz o mapeamento de dados, já foi visto em outras duas ocasiões (ver Figura 68 e

Figura 71), mas nessas, o mapeamento era mais simples, visto que apenas se relacionavam os campos origem dos campos destino. Neste caso, é necessário filtrar o registo correcto e ainda atribuir a um campo destino um valor gerado no momento do mapeamento (a data de conclusão). Na Figura 74 pode ser visto o mapeamento realizado para este caso em particular.

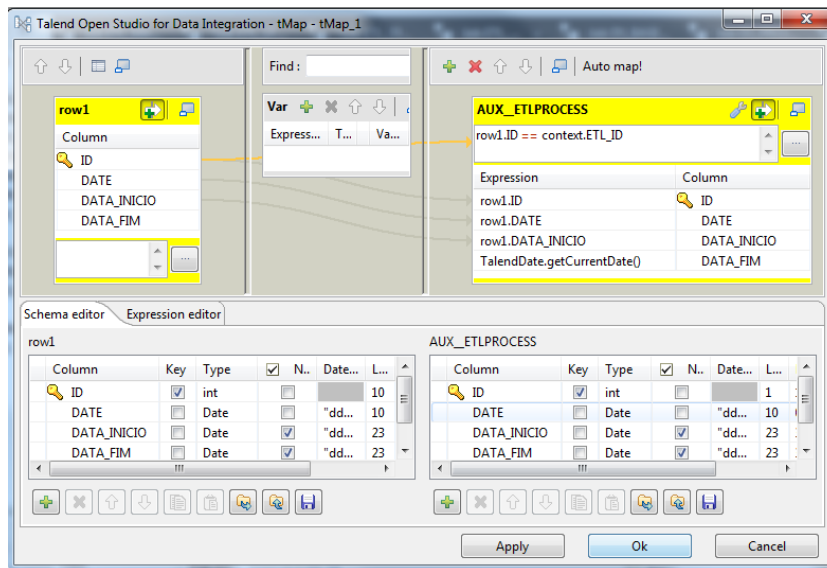


Figura 74 – Mapeamento de dados com atribuição de valor calculado e com filtragem de registo origem

Da fonte origem, foram escolhidos os campos *ID*, *DATE* E *DATA\_INICIO*. O registo para o qual se pretende fazer a actualização tem o *ID* que foi guardado, numa variável de contexto, no subprocesso *ETL\_START*. Na figura anterior, na tabela destino (tabela a amarelo mais à direita) foi definida a filtragem (neste caso, `row1.ID == context.ETL_ID`). Este filtro faz com que o *ID* do registo a actualizar na tabela destino seja igual à variável de contexto, que para este caso, tem o *ID* do registo inserido inicialmente. Outro aspecto que é importante de salientar, é que o campo *DATA\_FIM* deve ser calculado, de modo a que tenha o valor da data/hora no momento da actualização. Na figura anterior, no campo *DATA\_FIM* da tabela destino, foi inserida a expressão `TalendDate.getCurrentDate()`, esta expressão representa uma função do TOS que é chamada e retorna a data/hora actual. O terceiro componente representa a fonte destino dos dados mapeados, neste caso a tabela *AUX\_ETLPROCESS*. Este componente foi configurado, de modo a que a operação sobre os dados fosse uma operação de actualização. Na Figura 75 consegue-se perceber que na opção “Action on data” escolheu-se fazer a actualização dos dados (através do “Update”).

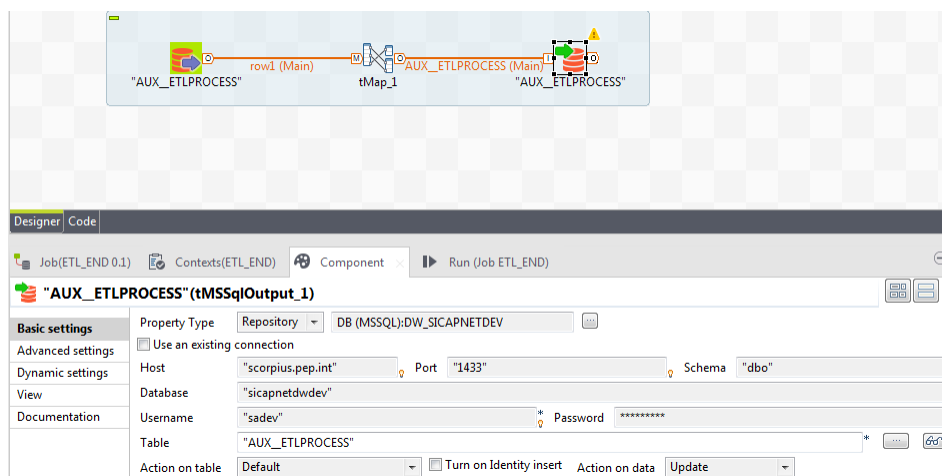


Figura 75 – Escolher a opção de actualização dos dados na fonte destino

### 5.3.4 Deployment dos Processos de ETL e Agendamento da Execução

Na secção anterior foram apresentados alguns subprocessos de ETL que formam o fluxo de dados que deve existir, diáriamente, entre o sistema fonte e o DW. Depois de criados os subprocessos é necessário fazer o *deployment* dos mesmos num servidor que trate da execução dos mesmos. Como já foi referido várias vezes neste documento, a ferramenta de BI escolhida foi o SpagoBI. Esta ferramenta consegue hospedar processos de ETL do TOS, tratando ainda da sua execução. Embora o SpagoBI tenha sido abordado, com mais detalhe, na secção anterior, é importante referir que é preciso instalá-lo de modo a carregar os processos de ETL a executar. Em [http://wiki.spagobi.org/xwiki/bin/export/spagobi\\_server/Installation\\_v3?format=pdf](http://wiki.spagobi.org/xwiki/bin/export/spagobi_server/Installation_v3?format=pdf) podem-se ver todos os passos seguidos para a instalação desta ferramenta.

O TOS consegue carregar um *job* directamente para um servidor que corra o SpagoBI. No entanto, esta configuração não vem configurada por omissão pelo TOS, sendo preciso configurar e activar o carregamento de *jobs* para o SpagoBI. Em <https://help.talend.com/display/TalendOpenStudioforDataIntegrationUserGuide54EN/5.7.1+How+to+deploy+a+Job+on+SpagoBI+server> podem ser consultados todos os passos para fazer esta configuração, assim como fazer o *deployment* dos *jobs* para o SpagoBI.

Depois de carregados os *jobs* para o SpagoBI, foi preciso configurar este de maneira a que corresse os *jobs* com uma periodicidade diária a uma determinada hora (a hora escolhida, e acordada com a equipa de BI da instituição, foi às 04h50m). Na Figura 76 pode-se ver que o *job* com nome *Yu\_Job* (que contém todos os subprocessos criados e referidos anteriormente) foi agendado de maneira a correr a partir de um de Outubro de 2015 às 04h50m.

## 5 – Criação do Data Warehouse e dos Processos de ETL

The screenshot shows the SpagoBI web interface in a browser window. The address bar indicates the URL is `localhost:8080/spagoBI/servlet/AdapterHTTP`. The interface is divided into several sections:

- Left Sidebar:** Contains navigation icons for Home, Back, Settings, Users, Folders, Reports, and Dashboards.
- Top Panel:** A search bar and a dropdown menu for 'Start Time' with an 'Add' button.
- Main Content Area:**
  - Overview Activity Yu\_Job:** A table with columns 'Label' and 'Description'. It lists 'AlarmInspectorJob', 'PROVA', and 'Yu\_Job'.
  - Documents:** A table with columns 'Name' and 'Parameters'. It lists 'Yu\_Job'.
  - Schedules:** A table with columns 'Name', 'Type', 'Start Date', 'Start Time', 'End Date', and 'End Time'. It lists 'Yu\_Job' with a type of 'day' and a start time of '04:50' on '01/10/2015'.

**Figura 76 – Agendamento do processo de ETL no SpagoBI**

## 6. Criação da Aplicação de Business Intelligence

Conhecida a aplicação transaccional existente, e depois de apresentado o DW criado, assim como os processos de ETL que extraíram e carregaram os dados do sistema fonte para o DW, chega o momento de apresentar a aplicação de BI criada. Esta aplicação teve como objectivo a aplicação prática dos conhecimentos adquiridos sobre BI, e a utilização da ferramenta de BI gratuita. Após a pesquisa e estudo das ferramentas, gratuitas, existentes no mercado, acredita-se que esta consegue responder a todos os requisitos apresentados pela organização a que se destina este trabalho, também como já foi referido anteriormente, a organização é a consultora Pessoas e Processos. No fim desta secção, poder-se-á avaliar a qualidade da solução que a ferramenta SpagoBI consegue fornecer aos seus utilizadores.

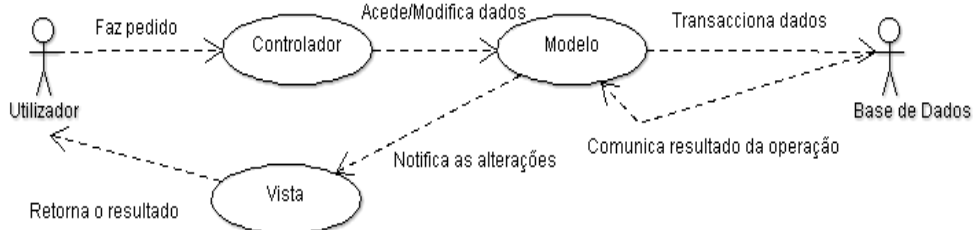
### 6.1 Tecnologias Utilizadas

Durante a criação da prova de conceito foram utilizadas várias tecnologias. A plataforma criada é apresentada aos utilizadores sob a forma de uma aplicação *web*. O Portal do Conhecimento, que é a plataforma de BI criada pela organização, tem este formato. Por isso, em conjunto com a equipa de BI da organização, acordou-se que este seria o formato criado. Por parte da organização não houve nenhuma limitação quanto às tecnologias a utilizar, tendo-se a liberdade de escolher as ferramentas que se considerassem serem as melhores.

#### 6.1.1 ASP MVC

A tecnologia escolhida, para criar a plataforma *web*, foi o Microsoft ASP MVC [95][96]. A escolha desta ferramenta deve-se essencialmente a dois motivos: o primeiro, e o mais importante, é que o padrão MVC (*Model, View, Controller*) [97], é actualmente o mais utilizado na criação de aplicações *web* (independentemente da tecnologia subjacente); o segundo, reside no facto desta tecnologia ser aquela que assegurará, dada a experiência e utilização anterior, uma maior rapidez, permitindo assim

produzir os mesmos resultados em menos tempo comparativamente com outras tecnologias (e.g.: ASP Web Forms, PHP e Java). É importante perceber o padrão MVC, sendo apresentada na Figura 77 a sua arquitectura.



**Figura 77 – Padrão MVC**

Como se pode constatar pela figura anterior, o padrão MVC divide-se em três camadas principais: o modelo, a vista e o controlador. O controlador é o grande responsável por receber os pedidos do utilizador. Quando um utilizador faz um pedido a uma página *web*, na verdade, com este padrão, está a fazer um pedido ao controlador. Este é responsável por receber os parâmetros do *Uniform Resource Locator* (URL), e por realizar a operação do pedido (e.g: selecção, actualização ou inserção ou eliminação de dados, entre outras operações). O controlador é também responsável por aceder ou modificar dados do modelo. O modelo é a camada que comunica com a base de dados onde os dados residem. Esta camada, pode também ser responsável por fazer os mapeamentos entre os objectos da base de dados e os da aplicação. O resultado é retornado na camada vista. Uma vista é a apresentação de uma página *web*, estando todos os componentes de uma página *web* na vista, este é o resultado final apresentado ao utilizador.

As principais vantagens do padrão MVC são [98]:

- Separação muito clara entre as camadas de visualização e as regras de negócios;
- A manutenção do sistema torna-se mais fácil;
- Permitir o reaproveitamento de código, principalmente da camada de modelo, que pode ser reutilizada em outros projetos;
- As alterações na camada de visualização não afectam as regras de negócios, já implementadas na camada do modelo;
- Permite o desenvolvimento, testes e manutenção de forma isolada entre as camadas.

Este padrão apresenta também algumas desvantagens face a outros existentes, são exemplo [98]:

- Em sistemas de baixa complexidade, este padrão pode criar uma complexidade desnecessária;
- Exige muita disciplina por parte dos programadores em relação à separação das camadas;
- Requer mais tempo para modelar o sistema;
- Este padrão, na sua implementação prática, requer mais tempo de aprendizagem.

### 6.1.2 Javascript

JavaScript [99][100] é uma linguagem de programação interpretada. Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e

interagissem com os utilizadores sem a necessidade dos scripts passarem pelo servidor. Através desta linguagem é possível gerir o navegador, realizar comunicação assíncrona e alterar o conteúdo de um documento *web*. É atualmente a principal linguagem para programação client-side em navegadores web. Foi concebida para ser uma linguagem script com orientação a objetos.

O SpagoBI é uma aplicação desenvolvida em linguagem java. Todos os componentes que são criados nesta ferramenta correm num servidor que interpreta esta linguagem. A integração dos componentes noutras tecnologias só é possível através de javascript. Como foi referido na secção 3.6.5, um dos módulos desta ferramenta é o o SpagoBI SDK. Este componente disponibiliza um ficheiro de javascript com funções que permitem integrar documentos existentes num servidor que corre o SpagoBI, em aplicações *web*. As linhas seguintes contêm código (do ficheiro disponibilizado pelo SpagoBI SDK), em javascript, que permite colocar um componente existente no SpagoBI numa página *web*.

```
<script type="text/javascript">

Sbi.sdk.services.setBaseUrl({
protocol: 'http'
, host: 'localhost'
, port: '8080'
, contextPath: 'SpagoBI'
, controllerPath: 'servlet/AdapterHTTP'
});

var cb = function(result, args, success) {

this.integrateComponent();
};

this.integrateComponent = function() {
Sbi.sdk.api.injectDocument({
documentLabel: 'DW_Tese_HChart_Par'
, executionRole: '/spagobi/admin'
, parameters: {department_id: 2}
, displayToolbar: false
, displaySliders: true
, target: 'grafico1'
, height: '450x'
, width: '100%'
, iframe: {
style: 'border: 0px;'
}
, useExtUI: true
});
};

</script>
```

Através do código javascript anterior é possível colocar, num elemento de uma página, um componente (*report*, gráfico, *dashboard*, consulta OLAP, etc.) criado e existente no SpagoBI. O exemplo anterior é um extracto de código existente na aplicação de BI criada. O objectivo do mesmo é

colocar num elemento da página, com *id* “gráfico1”, um gráfico criado no SpagoBI com o nome “DW\_Tese\_Hchart\_Par”. Esta operação pode ser vista na função *integrateComponent()*.

Através da linguagem javascript foi possível integrar os componentes, criados no SpagoBI, na aplicação de BI.

### 6.1.3 Estilos Responsivos

Foram utilizados estilos responsivos [101 no *layout* da aplicação criada. Até há alguns anos atrás o acesso a conteúdos *web* era feito usando um navegador num PC. Actualmente, este acesso é totalmente diferente. Os *smartphones* surgiram. Até mesmo os telemóveis mais simples têm navegadores *web* altamente eficazes e se não houver nenhum instalado, o utilizador pode facilmente descarregar um navegador. Além dos telemóveis e dos *smartphones*, existem os *tablets*, que permitem uma maior flexibilidade aos utilizadores que querem algo mais prático que um PC e mais confortável que um *smartphone*. Surge então a necessidade de criar aplicações e plataformas que se adaptem ao tamanho e resolução de vários ecrãs. O tamanho e resolução de um *smartphone* é bastante diferente ao dos *tablets* e PC’s. Os estilos responsivos surgem então para ajudar os programadores e *designers* a criar plataformas que se adaptem a resoluções diferentes.

A plataforma criada utilizou um *template* responsivo. Este é gratuito e *open source*, o que significa que o seu conteúdo pode ser completamente alterado e usado para uso comercial. O *template* pode ser encontrado em <https://almsaeedstudio.com/preview>. Aqui, pode ser descarregado o código-fonte e pode-se visualizar o resultado final num navegador, sendo ainda possível simular o resultado obtido através do acesso a partir de um telemóvel, *smartphone*, *tablet* ou PC. A Figura 78 mostra este *template*.

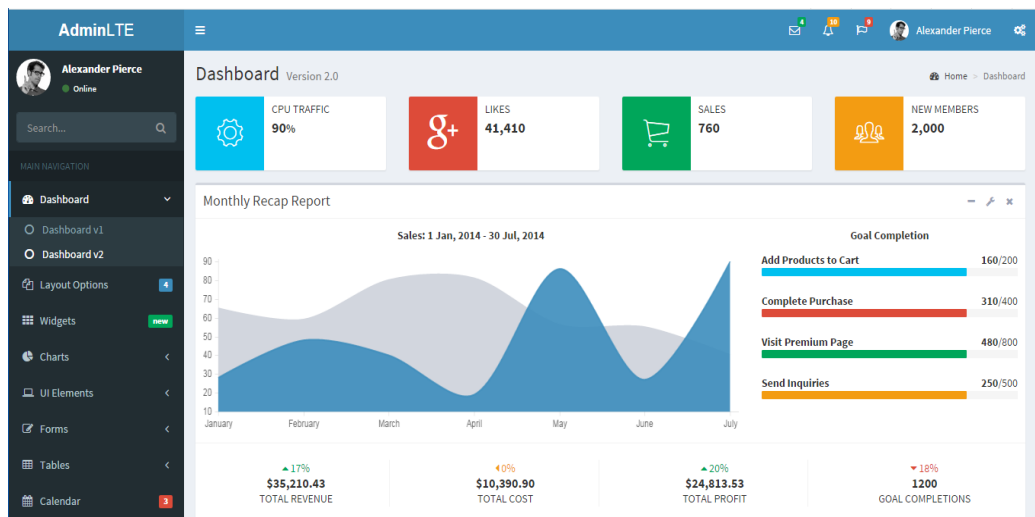


Figura 78 – Template responsivo utilizado na aplicação de BI criada

O conteúdo descarregado continha um conjunto de estilos e ficheiros javascript que estavam prontos a serem utilizados em qualquer tecnologia. No entanto, não existia nenhum projecto em ASP MVC que tivesse já configurados todos os estilos e ficheiros de javascript. Assim, criou-se um projecto ASP MVC já configurado que permite aos programadores utilizarem este *template*. Este projecto poupa tempo aos programadores, pois as configurações já estão feitas e prontas a serem utilizadas. No âmbito

deste trabalho disponibilizou-se o referido projeto gratuitamente, *online*, podendo ser encontrado e descarregado em <https://github.com/marquesdasilva/aspmvcbootstrapadminlte>. Para divulgar o projeto e dar conhecimento a programadores ASP MVC, foi criado um tópico num grupo da rede social LinkedIn. Este grupo foi criado por programadores desta tecnologia para discutir assuntos relacionados com esta. O tópico pode ser encontrado em <https://www.linkedin.com/grp/post/1931721-5973211846600065026>. O *feedback* e agradecimento de vários membros do grupo tem sido, até à data da escrita deste documento, bastante gratificante.

Com a utilização deste *template* responsivo, a aplicação de BI criada neste trabalho pode ser acedida por navegadores de telemóveis, *smartphones*, *tablets* e PC's. Com isto, o número de dispositivos e utilizadores alvo aumenta. A utilização deste *template* teve a aprovação da equipa de BI da organização.

## 6.2 Resultado Final – Aplicação de BI

Como já foi referido anteriormente, a aplicação de BI foi criada na tecnologia ASP MVC. Para criar e editar o código-fonte da aplicação foi usado o editor Visual Studio Express for Web 2013. O mesmo pode ser obtido em <https://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx>, e em <http://www.microsoft.com/en-us/download/details.aspx?id=44912> encontram-se os passos seguidos para a instalação do mesmo.

### 6.2.1 Página de Login

Os utilizadores podem aceder à aplicação de duas formas diferentes. A primeira é através de um atalho que existirá na aplicação transaccional. Através deste, o utilizador entra directamente na aplicação (sem precisar de *login*) pois já se encontra autenticado no primeiro sistema, e as suas credenciais serão automaticamente passadas para a aplicação de BI. A segunda forma é através da entrada directa numa das páginas da aplicação de BI. Em ambos os métodos, caso o utilizador não esteja autenticado, ou tenha feito *logout* da aplicação, o mesmo será redireccionado para a página de *login*. Esta página pode ser vista na Figura 79. A página de *login* é muito simples, consiste em dois campos de entrada (*username* e *password*), e para o utilizador entrar na aplicação deve usar as credenciais que usa no sistema transaccional.

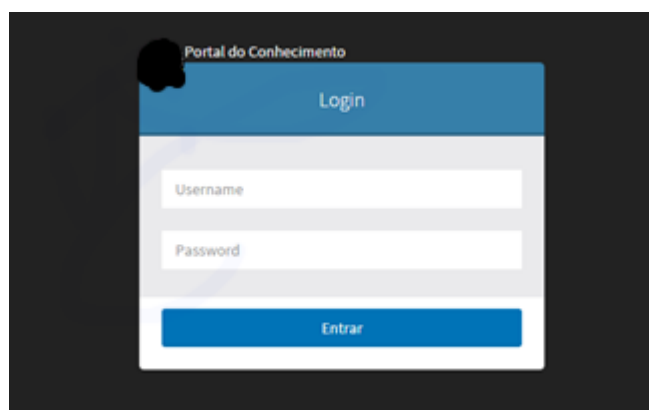


Figura 79 – Página de login

Na aplicação de BI desenvolvida não existe gestão de utilizadores. Os utilizadores são criados e geridos no aplicação transaccional. A única gestão de acessos que poderá existir, no futuro, na aplicação de BI, é a limitação a certos conteúdos. Se esta limitação vier a acontecer, consistirá na visualização de certos conteúdos, caso o utilizador tenha permissão para tal.

### 6.2.2 Organização da Aplicação

A aplicação de BI desenvolvida, utilizou então o *template* responsivo descrito na secção 6.1.3. Na Figura 80 consegue-se ver como esse *template* foi adaptado ao contexto da nova aplicação.

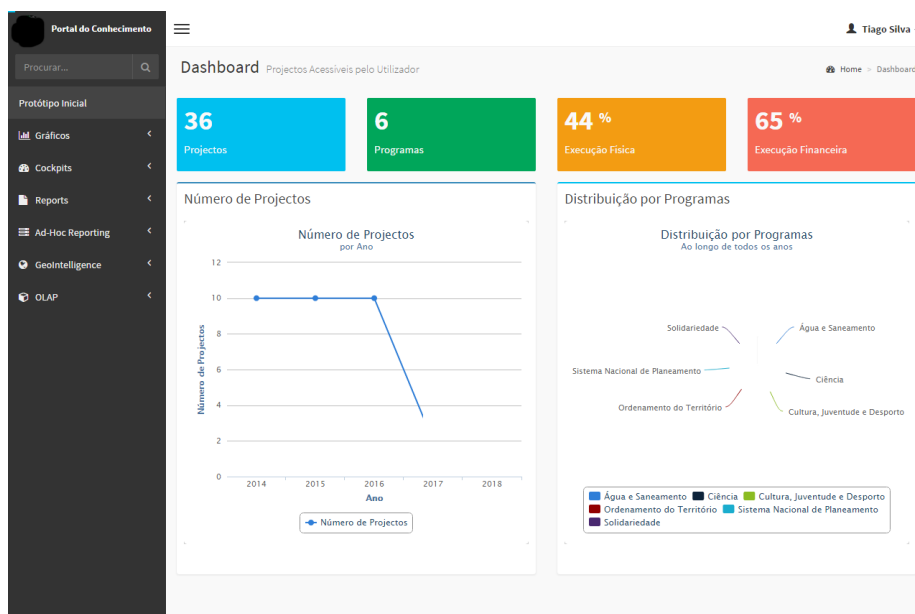


Figura 80 – Aplicação com template responsivo

A aplicação está dividida em três secções.

A primeira é o menu. Este pode-se ver no canto esquerdo da imagem anterior (secção com tons de cinzento escuro). Este menu divide-se em três partes:

A primeira parte é o logotipo e nome da aplicação. Na imagem anterior, o logotipo está censurado por motivos de confidencialidade. Quanto ao nome da aplicação, decidiu-se dar o mesmo nome da aplicação de BI já existente: Portal do Conhecimento.

A segunda parte do menu é constituído por uma caixa de pesquisa. O utilizador poderá navegar directamente para uma página da aplicação ao pesquisar por palavras-chaves, por exemplo, ao escrever “projecto” ou “projectos”, a aplicação abrirá a página resumo dos projectos.

A terceira parte do menu é constituída por um conjunto de ligações a partes da aplicação.

Na Figura 81 podem-se ver alguns itens do menu expandidos.

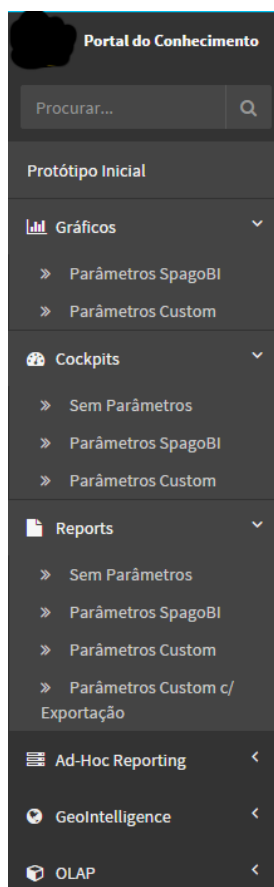


Figura 81 – Menu da aplicação

A segunda parte da aplicação é o menu de topo. Este menu, que pode ser visto na Figura 82, tem duas funções. A primeira é mostrar ou esconder o menu com as ligações às páginas (figura anterior). O utilizador, para aceder a esta funcionalidade, deve carregar na imagem com três linhas horizontais (parte superior esquerda da Figura 82). Escondendo o menu, a aplicação fica com mais espaço para o corpo da página (terceira parte da aplicação). A segunda função deste menu é permitir que o utilizador consiga ir para a página com o seu perfil (botão “Perfil”). Neste menu o utilizador poderá também sair da aplicação (botão “Sair”).

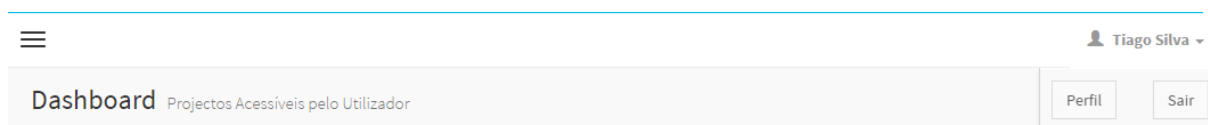


Figura 82 – Menu de topo

A terceira parte da aplicação é o corpo da página em si. Sempre que o utilizador acede a uma página diferente, o menu lateral (caso esteja visível) e o menu de topo mantêm-se iguais, o que muda é o corpo da página. Neste estão todos os conteúdos dentro do contexto a que o utilizador acede. Caso esteja a aceder a informação sobre projectos então o corpo da página terá, obviamente, informação sobre os projectos. Na Figura 83 pode-se ver um exemplo de página da aplicação de BI desenvolvida.

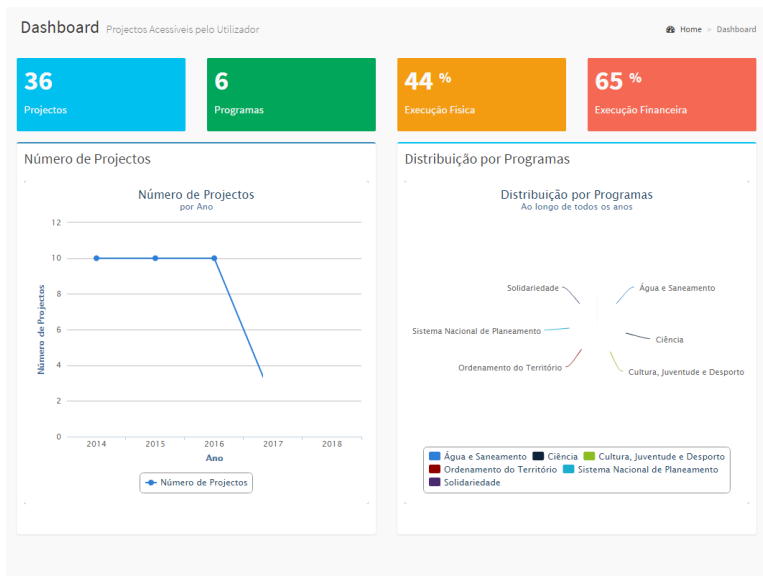


Figura 83 – Corpo de uma página

Por exclusão de partes, a parte da página que pode ser considerada o corpo da mesma, é a secção que não tem o menu lateral nem o menu de topo. Quando um utilizador muda de uma página para a outra, o conteúdo que muda é o corpo da página.

### 6.2.3 Página de Entrada

Quando um utilizador entra a primeira vez na aplicação, a página que aparece, por omissão, tem o nome de “Dashboard”. Esta página já foi mostrada anteriormente, ver Figura 80 e Figura 83. Esta página contém informação resumida sobre os projectos.

A primeira secção da página tem apenas indicadores numéricos, como se pode ver na Figura 84, esses indicadores são:

- Número total de projectos;
- Número de programas;
- Execução física dos projectos;
- Execução financeira dos projectos.



Figura 84 – Indicadores numéricos da página de entrada

Esta informação é obtida através de consultas directas ao DW. Para seleccionar o número de projectos é feita uma consulta à tabela de factos *FACT\_PROJECTO*, para consultar o número de programas é feita outra consulta à dimensão *DIM\_PROJ\_PROGRAMA*. As execuções dos projectos são calculadas, com regras de negócio internas. As tabelas de facto têm esta informação. As tabelas *FACT\_PROJECTO\_ACT\_FISIC\_EXEC* e a *FACT\_PROJECTO\_ACT\_FINANC\_EXEC* foram as tabelas consultadas, para respectivamente, retornarem o execução física e financeira dos projectos.

A segunda secção da página tem dois gráficos. Um mostra a evolução do número de projectos ao longo dos anos (Figura 85) e o outro tem a distribuição dos mesmos em programas (Figura 86).



Figura 85 – Gráfico com evolução do número de projectos ao longo do tempo



Figura 86 – Gráfico com distribuição dos projectos por programa

Estes dois gráficos foram construídos com o SpagoBI. O processo de criação dos mesmo já foi anteriormente mostrado.

### 6.2.4 Gráficos e Dashboards

Foram criadas várias páginas com informação relativas a projectos. Para todas as páginas da aplicação, foi requisito colocar filtros que permitissem que os utilizadores tivessem mais detalhe consoante

critérios de filtragem. Não serão mostradas, neste documento, todas as páginas com gráficos e *dashboards* desenvolvidas, por uma questão de objectividade (as páginas são todas muito idênticas no seu conteúdo, o que muda é a informação mostrada) e também de confidencialidade para com a organização. O exemplo que vai ser mostrado já foi visualizado anteriormente, consiste nos gráficos que podem ser visualizados na Figura 85 e na Figura 86. Estes gráficos foram escolhidos porque são simples de compreender, e não carecem de explicações sobre o tipo de informação que mostram.

Neste trabalho, para a parte dos gráficos e das *dashboards*, foram experimentadas, e aplicadas, as seguintes funcionalidades do SpagoBI:

- Utilização de gráficos independentes sem filtros;
- Utilização de gráficos independentes com filtros criados na ferramenta;
- Utilização de gráficos independentes com filtros externos;
- Dashboards sem filtros;
- Dashboards com filtros criados na ferramenta;
- Dashboards com filtros externos.

No SpagoBI podem-se criar gráficos com ou sem filtros. Um gráfico criado no SpagoBI, para ter filtros necessita que lhe sejam adicionados tipos de dados de filtragem, correspondentes aos filtros a aplicar. Opcionalmente, um gráfico pode ter filtros, mas o programador pode optar por não os mostrar.

O caso de um gráfico sem filtros já foi mostrado anteriormente. Na Figura 85 e na Figura 86, não são visualizados quaisquer filtros nos gráficos. Foram criados documentos, no SpagoBI, sem qualquer filtro.

Os componentes anteriores foram clonados, mas aos clones foram adicionados filtros e configurados os tipos de dados dos mesmos. Na Figura 87 podem-se ver os gráficos com filtros. Estes filtros são internos do SpagoBI. O filtro aplicado nos dois gráficos da figura seguinte, é o ano de análise. Para os dois casos foi escolhido o ano de 2017. Quando se filtra a informação para este ano, os dois gráficos actualizam automaticamente, mostrando para um caso, o número de projectos existentes em 2017 (gráfico mais à esquerda) e para o outro a distribuição do número de projectos nos programas existentes do ano em causa (gráfico mais à direita).

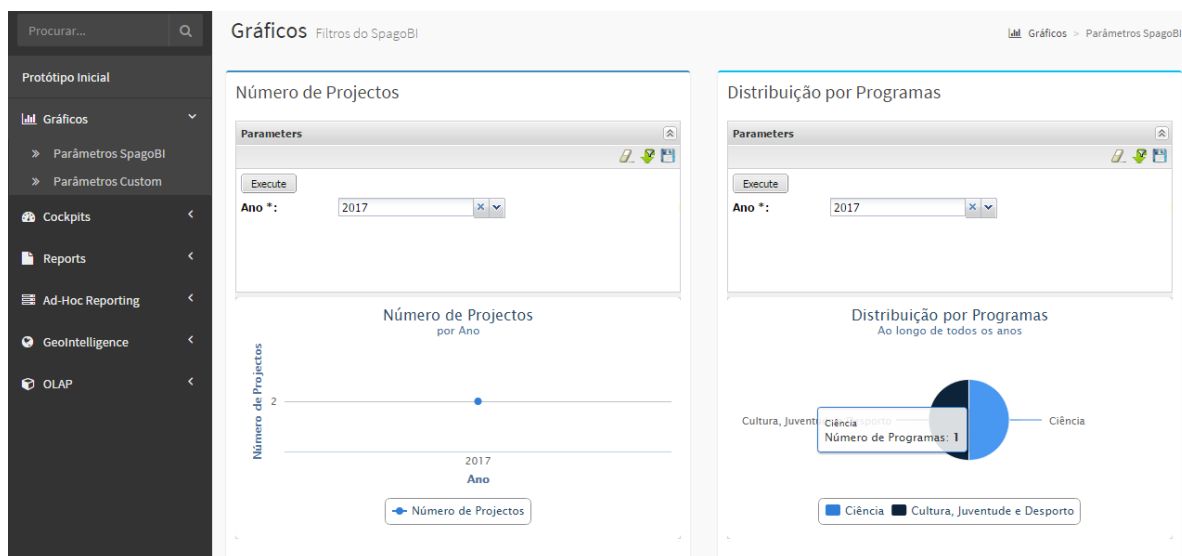


Figura 87 – Gráficos com filtros do SpagoBI

No SpagoBI, existe a opção de criar filtros nos componentes criados mas não mostrar os mesmos. Apesar de parecer contraditório, podem existir situações em que se deseja que o componente esteja preparado para receber parâmetros, mas que não sejam mostrados os filtros nas aplicações. Neste trabalho verificou-se que os filtros internos não são atractivos (do ponto de vista do *layout*) e a secção com os filtros ocupa muito espaço, diminuindo assim o tamanho dos gráficos. A utilização de filtros externos foi experimentada. Quando, neste trabalho, se refere um filtro como sendo externo, significa que o componente foi configurado de modo a receber parâmetros, mas a não mostrar os filtros internos. Isto significa que os componentes continuam a poder receber parâmetros, mas estes foram passados na chamada (utilizando javascript com o SpagoBI SDK) dos componentes. Na Figura 88 podem-se ver os gráficos, já mostrados anteriormente (ver Figura 85 e Figura 86), mas desta vez com filtros criados e desenhados na aplicação de BI.

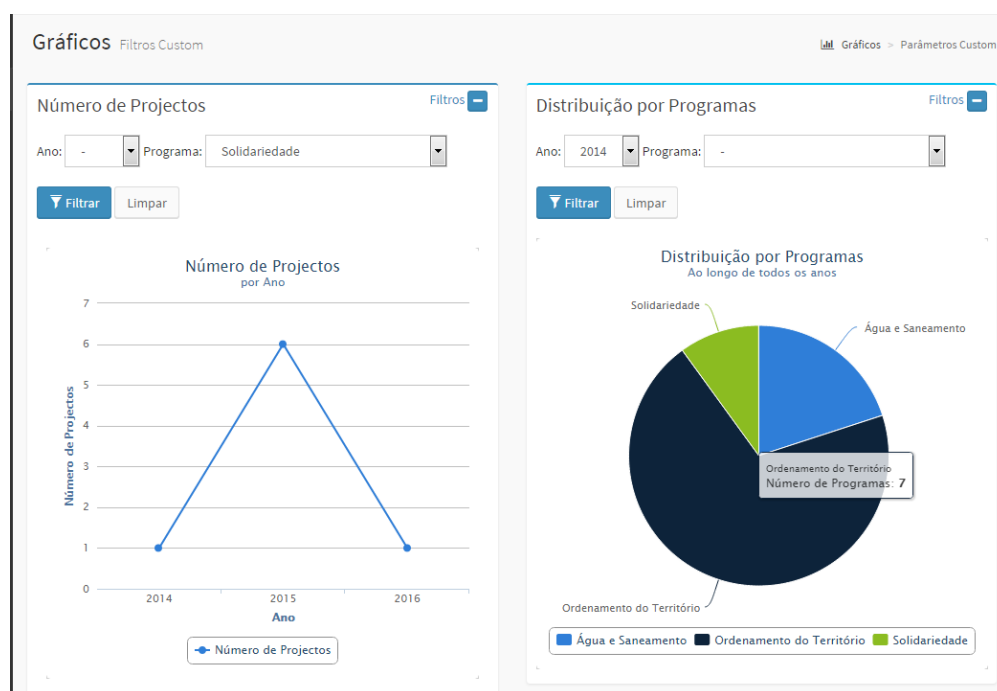


Figura 88 - Gráficos com filtros externos

Na figura anterior consegue-se ver que cada gráfico tem dois filtros, um é o ano e outro é o programa. No gráfico à esquerda, vê-se a evolução ao longo dos anos dos projectos inseridos no programa “Solidariedade”. No gráfico à direita, vê-se a distribuição do número de projectos por programa no ano de 2014. Com a utilização de filtros externos consegue-se um *layout* personalizado e ao gosto dos utilizadores. O tamanho dos gráficos é também maior quando comparado aos gráficos com filtros internos do SpagoBI.

Uma *dashboard*, no SpagoBI, é um aglomerado de componentes. Na Figura 89 pode-se ver um exemplo de uma *dashboard* sem filtros. Os gráficos anteriores foram usados, mais uma vez, para exemplificar esta funcionalidade. Foi criada uma *dashboard* composta por estes gráficos. Como se pode ver pela figura seguinte, os gráficos já não se encontram separados. Neste exemplo, eles fazem parte de um único documento, que é a *dashboard* em si.

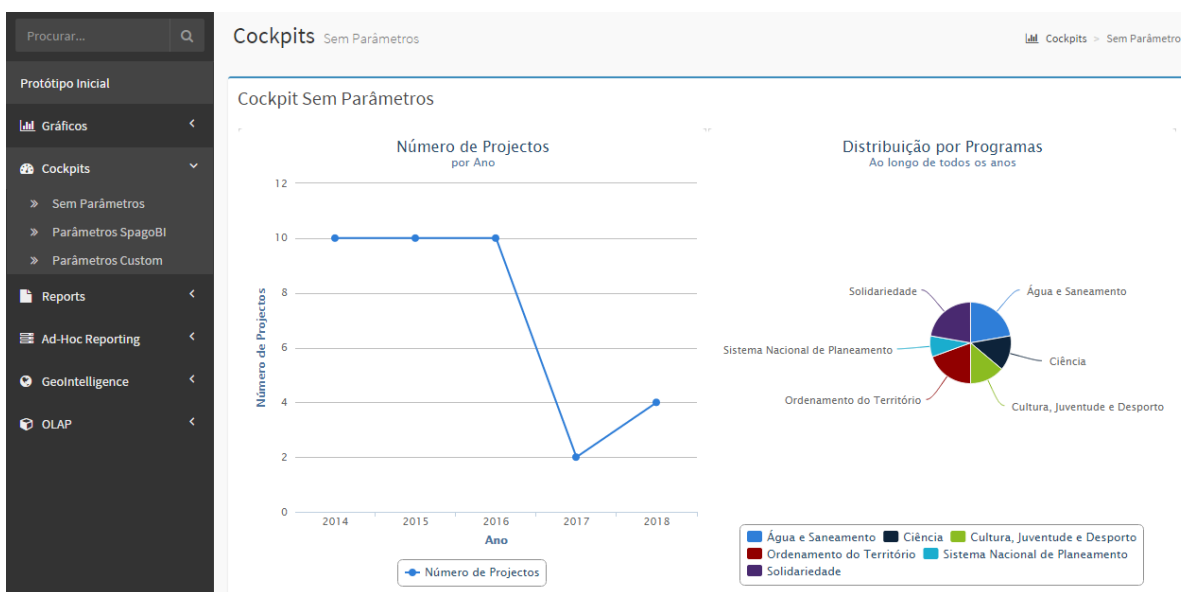


Figura 89 – Dashboard sem filtros

A criação de filtros nas *dashboards* segue o mesmo processo que a criação destes nos gráficos. Pode-se optar por criar *dashboards* com filtros internos (Figura 90) ou então podem-se configurar estas para receberem parâmetros, mas para não mostarem os filtros nas aplicações (Figura 91).

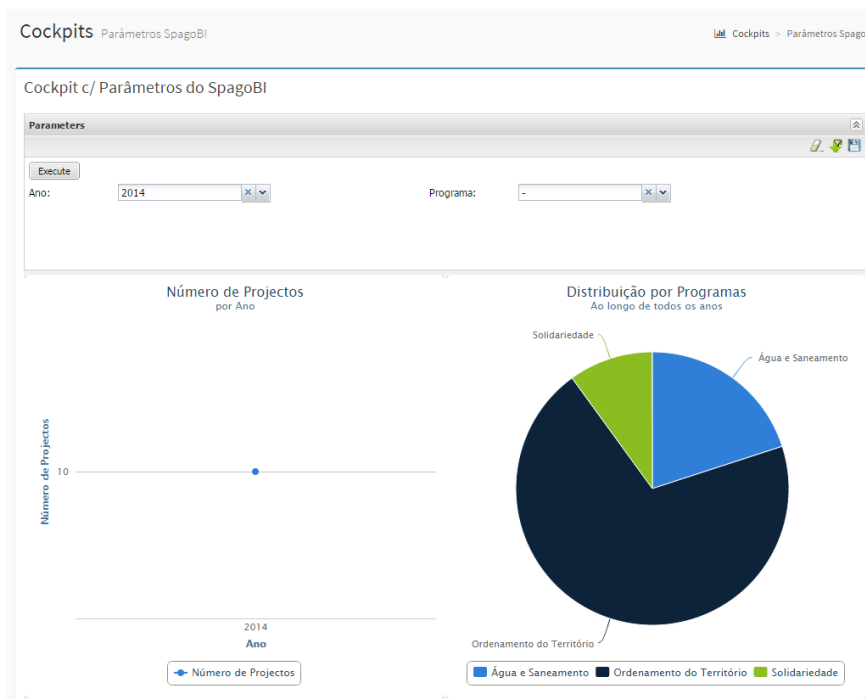


Figura 90 - Dashboard com filtros internos

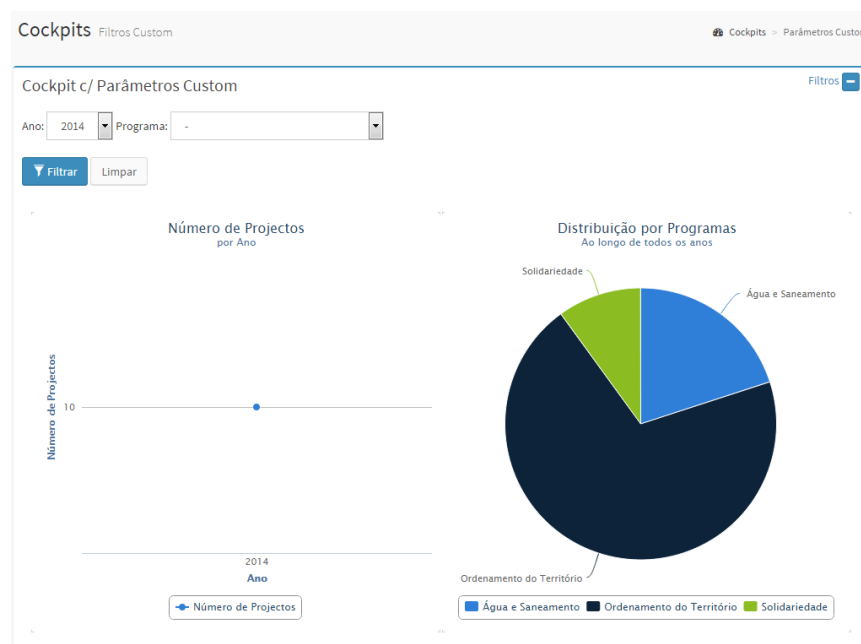


Figura 91 - Dashboard com filtros externos

Em ambas as *dashboards* foi aplicado o mesmo filtro. Foi escolhido o ano de 2014 como parâmetro de filtragem. Ao escolher o ano de 2014, ambos os gráficos foram automaticamente actualizados, de modo a mostrar a informação de acordo com o critério de filtragem. No SpagoBI, quando uma *dashboards* está configurada para receber parâmetros, ao existir uma filtragem, todos os componentes que formam a *dashboards* são automaticamente actualizados (desde que, obviamente, estejam também configurados com os mesmos parâmetros).

### 6.2.5 Reports

Foram criados diversos relatórios utilizando as funcionalidades de *reporting* do SpagoBI. O SpagoBI tem diversos motores de *reporting*. Como foi referido na secção 3.6.5, esta ferramenta utiliza os motores do JasperReports (secção JasperReports 3.3.1) e do BIRT Project (secção 3.3.2) para criar relatórios de dados. Foram criados relatórios com ambos os motores. No total, foram criados os seguintes relatórios:

- **Lista de projectos:** este relatório lista todos os projectos a que o utilizador tem acesso (dentro do seu departamento). Foram utilizados ambos os motores para criar os relatórios. Ambos os relatórios foram configurados para receber dois parâmetros. O primeiro é o ano de começo do projecto e o segundo é o programa à qual o projecto está associado;
- **Lista de programas:** este relatório lista todos os programas existentes no departamento do utilizador. Neste relatório, foram também utilizados os dois motores de *reporting*;
- **Lista com as execuções físicas dos projectos:** lista com as execuções físicas, em percentagem, dos projectos ao longo dos anos. Neste relatório aparecerem as execuções dos primeiros quatro anos dos projectos. Foi utilizado o motor do JasperReports para a criação deste relatório. Mais uma vez, os projectos que aparecem no relatório são os projectos a que o utilizador tem acesso, ou seja, projectos associados ao departamento do utilizador;
- **Lista com as execuções financeiras dos projectos:** lista bastante parecida com a anterior, ou seja, aparecem nesta as execuções financeiras dos primeiros quatro anos dos projectos. Foi utilizado o motor do BIRT Project para a criação deste relatório. Os projectos que aparecem nesta listagem, tal como nas anteriores, são aqueles a que o utilizador tem acesso.

Por uma questão de objectividade, não serão mostrados todos os relatórios criados. O aspecto dos relatórios, independentemente do motor utilizado, é também muito semelhante. Por estes motivos, optou-se por apenas mostrar o relatório com a listagem dos projectos. Este relatório pode ser visto na Figura 92. O formato do relatório, que aparece na figura seguinte, é o formato pdf mas foi também criado o formato excel. O relatório, em formato excel, pode ser visualizado na Figura 93. A importância de existir o formato excel, consiste no facto dos utilizadores poderem manipular, utilizando *software* para o efeito, os dados, podendo, por exemplo: criar filtros, fazer ordenações, etc. No SpagoBI não é preciso criar um componente diferente para cada formato do mesmo relatório. A ferramenta tem mecanismos próprios que fazem a exportação correcta, dependendo do formato escolhido. No trabalho desenvolvido apenas foram usados os formatos pdf e excel.

P&P Report

### Lista de Projectos

Código	Nome	Data de Inicio	Data de Fim	Programa
P2014.001	Projecto 1	Jan 30, 2014	Mar 30, 2014	Água e Saneamento - Descricao
P2014.002	Projecto 2	Jan 25, 2014	Mar 24, 2014	Água e Saneamento - Descricao
P2014.003	Projecto 3	May 5, 2014	Aug 24, 2014	Ordenamento do Território - Descricao
P2014.004	Projecto 4	Apr 8, 2014	Jun 24, 2014	Ordenamento do Território - Descricao
P2014.005	Projecto 5	Jun 18, 2014	Aug 2, 2014	Ordenamento do Território - Descricao
P2014.006	Projecto 6	Aug 24, 2014	Sep 2, 2014	Ordenamento do Território - Descricao
P2014.007	Projecto 7	Sep 2, 2014	Oct 2, 2014	Ordenamento do Território - Descricao
P2014.008	Projecto 8	Oct 4, 2014	Nov 6, 2014	Ordenamento do Território - Descricao
P2014.009	Projecto 9	Nov 1, 2014	Dec 9, 2014	Ordenamento do Território - Descricao
P2014.010	Projecto 10	Nov 15, 2014	Dec 28, 2014	Solidariedade - Descricao
P2015.001	Projecto 11	Jan 30, 2015	Mar 30, 2015	Solidariedade - Descricao
P2015.002	Projecto 12	Jan 28, 2015	Feb 25, 2015	Solidariedade - Descricao
P2015.003	Projecto 13	Feb 3, 2015	Mar 15, 2015	Água e Saneamento - Descricao
P2015.004	Projecto 14	Feb 20, 2015	Mar 25, 2015	Água e Saneamento - Descricao
P2015.005	Projecto 15	Mar 30, 2015	Jun 30, 2015	Solidariedade - Descricao
P2015.006	Projecto 16	Apr 28, 2015	May 25, 2015	Solidariedade - Descricao
P2015.007	Projecto 17	May 3, 2015	Jul 15, 2015	Água e Saneamento - Descricao
P2015.008	Projecto 18	Jun 20, 2015	Jul 25, 2015	Água e Saneamento - Descricao
P2015.009	Projecto 19	Aug 30, 2015	Oct 29, 2015	Solidariedade - Descricao
P2015.010	Projecto 20	Oct 28, 2015	Dec 25, 2015	Solidariedade - Descricao
P2016.001	Projecto 21	Jan 30, 2016	Mar 30, 2016	Cultura, Juventude e Desporto - Descricao
P2016.002	Projecto 22	Jan 28, 2016	Feb 25, 2016	Ciência - Descricao
P2016.003	Projecto 23	Feb 3, 2016	Mar 15, 2016	Sistema Nacional de Planeamento - Descricao
P2016.004	Projecto 24	Feb 20, 2016	Mar 25, 2016	Sistema Nacional de Planeamento - Descricao
P2016.005	Projecto 25	Mar 30, 2016	Jun 30, 2016	Sistema Nacional de Planeamento - Descricao
P2016.006	Projecto 26	Apr 28, 2016	May 25, 2016	Solidariedade - Descricao
P2016.007	Projecto 27	May 3, 2016	Jul 15, 2016	Água e Saneamento - Descricao
P2016.008	Projecto 28	Jun 20, 2016	Jul 25, 2016	Água e Saneamento - Descricao
P2016.009	Projecto 29	Aug 30, 2016	Oct 29, 2016	Ciência - Descricao
P2016.010	Projecto 30	Oct 28, 2016	Dec 25, 2016	Cultura, Juventude e Desporto - Descricao
P2017.001	Projecto 31	Jan 30, 2017	Mar 30, 2016	Cultura, Juventude e Desporto - Descricao
P2017.002	Projecto 32	Apr 28, 2017	May 25, 2016	Ciência - Descricao

Figura 92 – Relatório com a lista de projectos (formato pdf)

The image shows the Microsoft Excel interface with the same project list data from Figure 92. The ribbon includes FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, DATA, REVIEW, VIEW, DEVELOPER, and TEAM. The spreadsheet grid shows columns A through T and rows 1 through 117. The data is organized into a table with headers: Código, Nome, Data de Inicio, Data de Fim, and Programa. The content of the table is identical to the one in Figure 92.

Figura 93 - Relatório com a lista de projectos (formato excel)

Os relatórios criados com o SpagoBI podem ser integrados nas aplicações externas de duas formas. A primeira é colocar o relatório embutido na aplicação. Um relatório embutido significa que o mesmo existe, e pode sofrer interacções por parte dos utilizadores nas aplicações. A segunda forma de utilizar

## 6 – Criação da Aplicação de Business Intelligence

um relatório consiste, simplesmente, na sua exportação. Estas duas formas podem ser combinadas, isto é, embutir um relatório na aplicação e permitir que o mesmo seja exportado através de uma outra chamada ao documento por parte da aplicação. Na Figura 94 pode-se ver o relatório da lista de projectos embutido na aplicação. Na Figura 95 está representado um caso em que não foi embutido um relatório, mas é possível exportar o mesmo para os formatos pdf e excel.

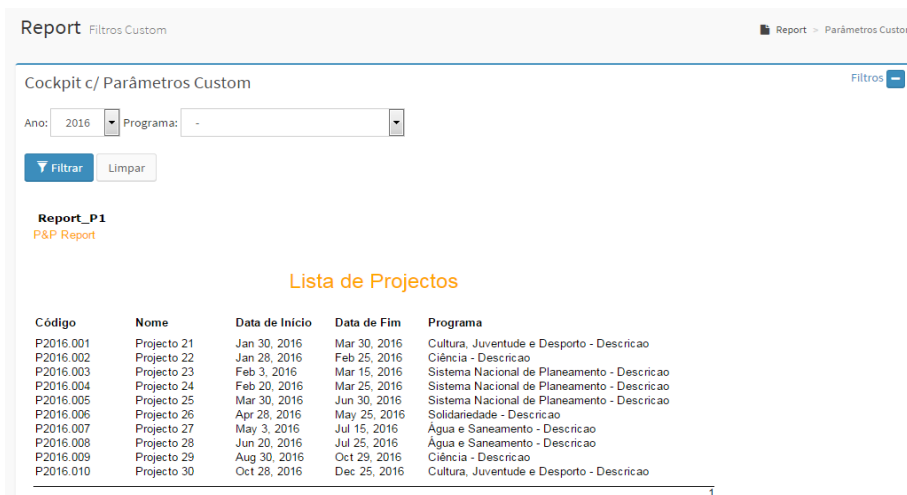


Figura 94 – Relatório embutido na aplicação com parâmetros externos

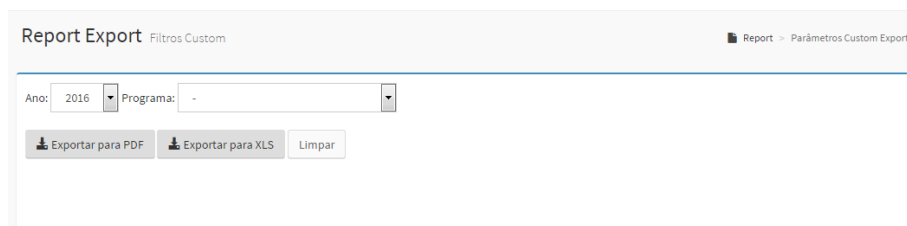


Figura 95 – Exportação de relatório com parâmetros externos

Tal como nos gráficos e nas *dashboards* podem-se configurar os relatórios para terem filtros internos e externos. Foram experimentados ambos os casos. Nas figuras anteriores podem-se ver os filtros criados. Em ambos os casos, os filtros são externos. Na Figura 96 pode-se ver o mesmo relatório com filtros internos.

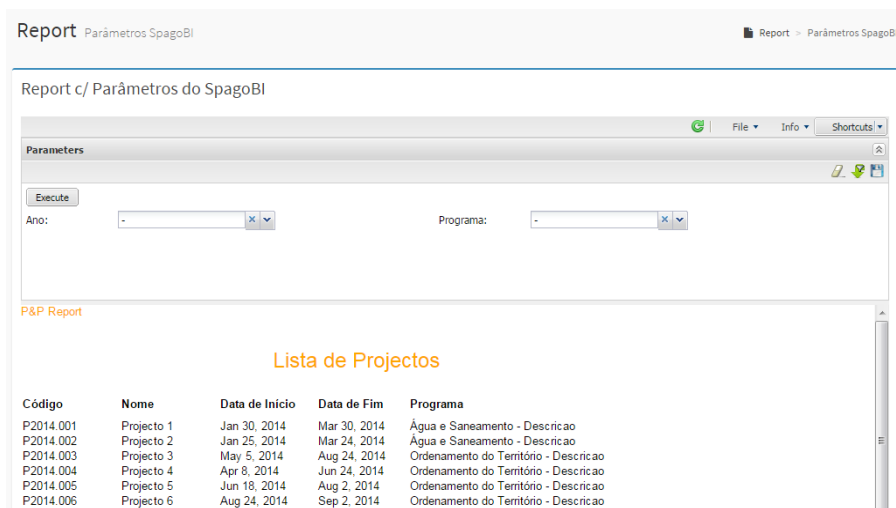


Figura 96 – Relatório embutido com parâmetros internos

### 6.2.6 Informação Geográfica

Embora não fosse objectivo da aplicação fornecer informação geográfica, a experimentação deste módulo revelou uma qualidade na solução fornecida assinalável. Juntamente com a equipa de BI, decidiu-se aprofundar esta área. Como resultado, criou-se um componente com informação geográfica dos projectos. Por motivos de confidencialidade, não será mostrado o componente experimental criado. No entanto, o componente é muito idêntico ao que pode ser visualizado na Figura 97.

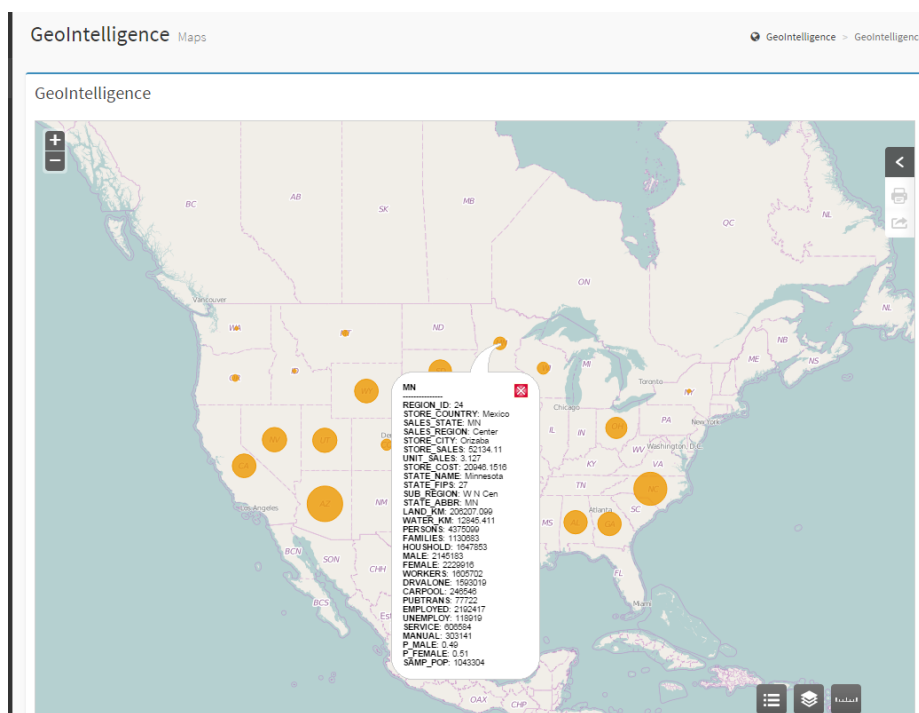


Figura 97 – Exemplo de componente com informação geográfica

Tal como o componente que pode ser visto na figura anterior, o componente criado tem informação distribuída geograficamente. Na figura anterior consegue-se perceber que é mostrada informação, neste caso sobre vendas de produtos, em cada estado dos Estados Unidos da América. Cada estado tem um foco cor de laranja, que aumenta, consoante o volume de vendas existente. Ao seleccionar um estado é aberta uma janela com informação mais detalhada. O componente criado é muito parecido. Nesse é mostrado o país do cliente (único país onde tem projectos associados ao momento da criação deste documento) separado por distritos. Cada distrito tem um foco, tal como na figura anterior, cor de laranja, em que o mesmo é maior quanto o número de projectos existentes nesse distrito. Ao carregar sobre o distrito, são mostradas informações detalhadas sobre os projectos. Esta informação detalhada consiste no somatório do valor planeado dos projectos para todos os anos, assim como a média das execuções físicas e financeiras enúmero de programas.

### 6.3 Outros Aspectos

Ao longo das secções anteriores foram listadas as tecnologias que foram utilizadas e os resultados obtidos. Os resultados demonstrados fazem parte de um leque alargado de conteúdos criados, e supervisionados pela equipa de BI da instituição. Por motivos de confidencialidade, não foram

mostrados todos os componentes, nem se entrou em detalhes sobre a natureza dos dados utilizados para análise. No entanto, foi feito um esforço para que os conteúdos mostrados anteriormente fossem suficientes para mostrar, em primeiro lugar, o que foi feito de mais relevante, e em segundo lugar, a qualidade que os mesmos apresentam. É também importante referir, que alguns dos conteúdos apresentados, não são a versão final que será disponibilizada. O objectivo deste capítulo, e da aplicação de BI em geral, ao momento da criação deste documento, é mostrar que a ferramenta escolhida é capaz de produzir resultados práticos com qualidade suficiente para serem distribuídos num produto comercial. Existem também outras áreas de BI, que estão a ser estudadas, mas que devido à imaturidade das soluções até ao momento desenvolvidas, não foram apresentadas. Duas dessas áreas são os relatórios *ad-hoc* e as consultas OLAP.

## 7. Conclusões e Trabalho Futuro

A importância dos sistemas de gestão nas organizações é, já à alguns anos, evidente. As empresas precisam de guardar todos os dados dos seus processos de negócio. Estes são, na maioria das vezes, guardados em bases de dados operacionais. Estas bases de dados têm como objectivo guardar os dados actuais dos processos das organizações. Existem vários sistemas que guardam e gerem estes dados. Actualmente, o uso de ERP's, CRM's, SCM's e outros sistemas de gestão é comum nas organizações. Mas os dados por si só, não têm grande significado: é preciso traduzir estes em informação concreta sobre o negócio a que dizem respeito. Se até há alguns anos atrás, ter a informação sobre os processos de negócio chegava para a gestão das organizações, nos dias que correm esta informação, apesar de importante, não apresenta nenhuma vantagem competitiva à organização. É apenas informação.

No capítulo de introdução deste trabalho, foi referido que nos últimos anos têm existido factores de instabilidade económica. As empresas perderam poder de compra, e sentiram a necessidade de fazerem esforços para reduzir as suas despesas e custos relacionados com as actividades que executam. Noutras palavras, as organizações precisaram de otimizar os seus processos de negócio. A procura por diferenciação e competitividade comercial aumentou com as dificuldades económicas dos últimos anos. As organizações começam cada vez mais a reflectir sobre a importância da informação que possuem. Se, no passado, houve a necessidade de transformar os dados em informação, nos últimos anos, houve a consciencialização que é preciso algo mais, é necessário um nova etapa neste processo. Esta etapa, já anteriormente referida, é a transformação da informação em conhecimento. A existência de conhecimento, através da informação que as organizações já possuem, ajuda a tomada de decisões. Estas decisões depois fazem com que as empresas possam ganhar competitividade económica e, mais importante ainda, se consigam diferenciar da concorrência, através da melhoria e optimização dos processos de negócio inerentes às actividades de mercado que praticam. Neste ambiente surge a *business intelligence*.

Como também já foi referido, a *business intelligence* tem como principal objectivo a recolha de informação que possa ser usada para tomar decisões. Existem muitas ferramentas que permitem criar aplicações de BI. No entanto, a maioria delas tem custos de licenças associados, que, na grande

maioria dos casos, são elevados. Se as empresas precisam de reduzir as suas despesas e custos inerentes às actividades que praticam, a utilização destas ferramentas pode tornar-se inviável face ao custo das mesmas. Este problema acentua-se nas pequenas e médias empresas, que muito provavelmente, não têm ao seu dispor as condições económicas necessárias para adquirir estas licenças. No entanto, actualmente existem soluções gratuitas que conseguem dar resultados com qualidade suficiente e que justifica a sua adopção.

O grande objectivo deste trabalho foi estudar a viabilidade das opções existentes no mercado, no que toca às soluções gratuitas, para uso comercial, na área da *business intelligence*. Foram apresentadas, na secção 3, todas as ferramentas estudadas e experimentadas. Depois de referidas as opções existentes no mercado, conseguiu-se chegar a uma primeira conclusão. Existem, em número, e funcionalidades, várias ferramentas no mercado que, gratuitamente, permitem as organizações criar aplicações de BI. Não é por escassez de alternativas que as organizações não optam por estas soluções. Existem ferramentas para diversas áreas, as áreas estudadas foram: ETL/data integration; *reporting*, *data mining*, *dashboarding* e OLAP. Também é importante de analisar, e esta análise foi outro objectivo deste trabalho, se estas opções conseguem oferecer qualidade suficiente para serem utilizadas no dia-a-dia das organizações ou na criação de produtos comerciais (como foi o caso). Neste trabalho, em colaboração com a equipa de BI da organização Pessoas e Processos, foram estipulados alguns requisitos para a escolha de uma ou várias ferramentas para serem alvo de uma análise mais aprofundada. Depois de uma análise comparativa cuidada, e que foi exposta na secção 3, e de acordo com os requisitos estipulados, chegou-se à conclusão que a ferramenta SpagoBI merecia esta análise e experimentação mais cuidada.

A organização Pessoas e Processos expôs um sistema existente, e que podia ser utilizado, para a prova de conceitos prática da ferramenta escolhida. Em colaboração com a equipa de BI da organização, foi criado um DW (e feitos processos de ETL) que permitam a disponibilização de dados que pudessem ser usados nas análises feitas na plataforma de BI criada. Nesta fase houve a experimentação da ferramenta TOS. Esta é a ferramenta utilizada pelo SpagoBI para criar os processos de ETL e de integração de dados. Depois da sua utilização prática, na criação dos processos de ETL necessários, chegou-se à conclusão que esta é uma alternativa bastante viável para as organizações utilizarem na área da integração de dados. A criação dos processos de ETL revelou-se desafiante, mas este desafio foi cumprido com alguma serenidade, com a utilização do TOS. Esta ferramenta (e o seu editor) é simples de utilizar e existe muita documentação que ajuda os utilizadores a criarem os processos de ETL. O TOS, seja utilizado independentemente, ou com outra ferramenta (neste caso o SpagoBI), deve ser uma alternativa para as organizações. Estas devem averiguar se o TOS consegue dar resposta às necessidades existentes. Caso consiga dar resposta às suas necessidades, as empresas podem estar a desperdiçar a oportunidade de terem uma ferramenta completa, simples e eficaz, sem qualquer custo de licenciamento.

Depois de criado o DW e os processos de ETL necessários para transferir a informação do sistema transaccional para o DW e respetiva execução, experimentou-se, com detalhe e profundidade, a ferramenta SpagoBI. Esta foi utilizada para criar um conjunto de componentes de BI, cuja qualidade final e a avaliação do próprio processo de criação, determinassem se esta ferramenta seria uma opção viável para a criação de um produto comercial. No capítulo 6 foi demonstrada a aplicação de BI criada. A criação dos componentes no SpagoBI foi, mais uma vez, um desafio, mas realmente,

verificou-se que a criação dos gráficos e *dashboards* se revelou uma tarefa simples de fazer. Nesta área, a primeira dificuldade foi a configuração destes componentes para receberem parâmetros. Esta dificuldade foi ultrapassada através dos manuais de ajuda da ferramenta e de tópicos existentes no fórum de programadores do SpagoBI. Nesta fase, ficou provado que existe documentação suficiente para responder a problemas comuns, como foi o caso da configuração dos parâmetros.

A área de *reporting* também se revelou interessante. O SpagoBI tem os dois motores *open source* mais usados (e também já referidos neste trabalho, ver as secções 3.3.1 e 3.3.2), isto faz com que, dentro da mesma área (neste caso a área de *reporting*), existam duas alternativas e duas formas para criar relatórios. A criação de relatórios revelou-se uma tarefa demorada. Os dois motores de *reporting*, apesar de muito parecidos, têm mecanismos diferentes para, por exemplo, configurar as fontes de dados, criar áreas dentro do relatório, criar filtros dentro do *report*, etc. O tempo dedicado à criação dos relatórios foi superior em comparação ao tempo dedicado às outras áreas. Verificou-se que é preciso algum tempo de aprendizagem para os utilizadores começarem a criar os relatórios. No entanto, ao fim desta fase de aprendizagem, o processo de criação, pela sua repetitividade, torna-se acessível.

A área da informação geográfica foi uma agradável surpresa. Como já foi referido anteriormente, inicialmente não existiam planos para aprofundar esta vertente. No entanto, após uma breve experimentação e demonstração dos resultados obtidos, a equipa de BI deu luz verde para se aprofundar esta área. O componente criado revelou uma qualidade impressionante. O bom desempenho e o agradável aspecto gráfico final foram notórios. O processo de criação deste componente revelou-se algo moroso, pois foi preciso configurar muitos parâmetros e testar os resultados mostrados no gráfico espacial. No entanto, apesar de moroso, a criação deste componente revelou-se simples. Foram criados outros componentes, e aprofundadas outras áreas de BI. No entanto, à data da finalização deste documento, o trabalho desenvolvido nestas áreas (relatórios *ad-hoc* e consultas OLAP), ainda não era conclusivo. Por isso, optou-se por não apresentar o trabalho criado. Em suma, depois de criados vários componentes e de aprofundadas várias áreas de BI, o SpagoBI revelou-se uma ferramenta capaz e com produção de resultados com qualidade suficiente para que esta ferramenta seja usada no dia-a-dia das organizações ou na criação de produtos de BI comerciais.

Para resumir, com este trabalho conseguiram-se obter as seguintes conclusões:

- É cada vez mais importante, nos dias que correm, as organizações ganharem mais competitividade;
- Apesar da importância da informação que as organizações possuem, é preciso transformar esta em conhecimento. Este conhecimento deverá apresentar-se como algo diferenciador, devendo ainda otimizar os processos de negócios das empresas;
- A *business intelligence* apresenta-se como um instrumento importante no que toca à transformação da informação em conhecimentos, que podem trazer às organizações vantagens competitivas. A BI é um instrumento poderoso que deve ser aplicada de maneira a que consiga dar uma visão do negócio aos gestores que permita que estes consigam tomar as melhores decisões de negócio;

- O custo das ferramentas de BI comerciais pode obstar à utilização destas em empresas que tenham a necessidade de reduzir as suas despesas e custos tecnológicos. Este impedimento aguniza-se em pequenas e médias empresas, em regra, com poucos recursos financeiros. No entanto, existem ferramentas gratuitas que oferecem qualidade suficiente nas soluções oferecidas;
- A utilização de ferramentas gratuitas de BI deve ser reflectido. As organizações devem fazer um levantamento das necessidades existentes, e posteriormente estabelecer requisitos que devem ser cumpridos pelas ferramentas a utilizar;
- Existem muitas alternativas no mercado no que toca a ferramentas gratuitas de BI. As áreas de BI abrangidas por estas ferramentas vão desde a área de *reporting* até às consultas OLAP, passando por *data mining*, *dashboarding*, etc.;
- As ferramentas do tipo *suite* apresentam-se como fortes candidatas à adopção destas nas organizações. Comparativamente com ferramentas isoladas, as ferramentas *suite* apresentam uma solução integrada que responde a diversas áreas da BI;
- A prova de conceitos prática, que consistiu na criação de um DW, processos de ETL e aplicação de BI revelou que estas ferramentas são viáveis para serem utilizadas nas organizações. Tanto o TOS como o SpagoBI merecem atenção por parte das organizações. Estas ferramentas conseguiram produzir resultados de qualidade.

As conclusões listadas anteriormente, são as principais a retirar deste trabalho. Como já foi referido, a importância de fazer *business intelligence* é enorme. A utilização de ferramentas gratuitas é viável e as organizações devem fazer um esforço para, dependendo dos seus requisitos, estudar a utilização destas nas suas soluções internas e externas.

Todo o trabalho desenvolvido foi um grande desafio pessoal e profissional. O tempo de estudo e experimentação das várias ferramentas apresentadas, foi desgastante, mas ao mesmo tempo conseguiu-se, com este trabalho, provar que existem muitas alternativas no mercado para as organizações poderem adoptar. Com este trabalho, ganhou-se competências na área de *data warehousing*, *data integration*, e como é claro, na área de *business intelligence*.

No entanto, existem ainda muitos desenvolvimentos práticos para realizar. Fica para trabalho futuro finalizar o desenvolvimento da plataforma oficial, experimentar outras funcionalidades do SpagoBI (e.g.: OLAP, *balanced scorecard*, etc.) e se possível analisar ainda outras ferramentas, do mesmo tipo, que possam complementar o SpagoBI nas mais diversas áreas.

## REFERÊNCIAS

- [1] Coker F. (2014). *Understanding the Vital Signs of Your Business*. Ambient Light Publishing. pp. 41–42
- [2] Miller D., Richard C. (2001). *Cyclopaedia of Commercial and Business Anecdotes; Comprising Interesting Reminiscences and Facts, Remarkable Traits and Humors of Merchants, Traders, Bankers Etc. in All Ages and Countries*. D. Appleton and company pp. 210
- [3] Luhn H. (1958). *A Business Intelligence System*. IBM Journal
- [4] Power D. (2007). *A Brief History of Decision Support Systems, version 4.0*. DSSResources
- [5] Power D.. *A Brief History of Decision Support Systems*
- [6] Anandarajan M, Anandarajan A, Srinivasan CA. (2014). *Business Intelligence Techniques: A Perspective from Accounting and Finance*. Springer, Berlim
- [7] Software Advice. *Compare Business Intelligence (BI) Software Tools*
- [8] European Commission (2009). *Economic Crisis in Europe: Causes, Consequences and Responses*. European Economy 7
- [9] McKibbin W., Stoeckel A. (2009). *The Global Financial Crisis: Causes and Consequences*. Working Papers in Internacional Economics
- [10] Kobiellus J. (2010). *What's Not BI? Oh, Don't Get Me Started....Oops Too Late...Here Goes....*
- [11] Weber S. (2004). *The Success of Open Source*. Cambridge, MA: Harvard University Press. pp. 38–44
- [12] Muffatto M. (2006). *Open Source: A Multidisciplinary Approach*. Imperial College Press
- [13] Campbell T., Stanley C. (1963). *Experimental and Quasiexperimental Designs for Research*. Chicago, IL: Rand McNally
- [14] Kirk E. (1995). *Experimental Design: Procedures for the Behavioral Sciences*. Pacific Grove, CA: Brooks/Cole
- [15] Burt C. (1954). *The Differentiation Of Intellectual Ability*. The British Journal of Educational Psychology
- [16] Sawyer D. (1994). *The Art of War*. Westview Press
- [17] Escorel L. (1979). *Introdução ao Pensamento Político de Maquiavel*. Universidade de Brasília paper 19
- [18] Humphrey A. (2005). *SWOT Analysis for Management Consulting*. SRI Alumni Newsletter
- [19] Marvin S. (1985). *Personal Computers and the Family*. Routledge, 1985
- [20] Ralston A., Reilly E. (1993). *Workstation*. Encyclopedia of Computer Science, Nova Iorque
- [21] Rouse M. (2015). *ERP (Enterprise Resource Planning)*. TechTarget
- [22] King. W. (2005), *Ensuring ERP Implementation Success*. Information Systems Management
- [23] Yun Z., Joseph W., David C. (2003). *Customer Relationship Management (CRM) in Business-to-Business (B2B) E-commerce*. Emerald 11

- [24] Bolton N. (1998). *A Dynamic Model of the Duration of the Customer's Relationship with a Continuous Service Provider: The Role of Satisfaction*. Marketing Science
- [25] Bartsch F. (2013). *Supply Chain Management (SCM)*. BB Handel
- [26] Inmon B. (1992). *Building the Data Warehouse*. Wiley
- [27] Kimball R. (1996). *The Data Warehouse Toolkit*. Wiley
- [28] Porter E. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*. Nova Iorque
- [29] Watson J., Barbara H. (2007). *The Current State of Business Intelligence*
- [30] Andrew B. (2013). *Gartner releases 2013 BI Magic Quadrant*. ZDNet
- [31] Swain S. (2008). *Business Intelligence for Dummies*
- [32] Rodrigues L. (2006). *Business Intelligence*
- [33] Golfarelli/Rizzi, *Data Warehouse Design*, pp. 291
- [34] Oracle (2007). *Overview of Extraction, Transformation, and Loading*, Database Data Warehousing Guide
- [35] Polsku P. (2007). *ETL Process*. Datawarehouse4u
- [36] Baamann K. (2002). *Data Quality Aspects of Revenue Assurance*, pp. 3-5
- [37] Eckerson W. (2002). *Data Warehousing Special Report: Data Quality and the Bottom Line*
- [38] Lymer A. (2013). *Business Reporting on Internet*
- [39] Hill G. (2013). *A Guide to Enterprise Reporting*
- [40] Inmon B. (2013). *Operational and Informational Reporting*. Information Management
- [41] McFadden P. (2012). *What is Dashboard Reporting*
- [42] Briggs J. (2013). *Management Reports & Dashboard Best Practice*. Target Dashboard
- [43] Gibbon C. (1990). *Performance Indicators*, BERA Dialogues
- [44] Kaplan S., Norton P. (1992). *The Balanced Scorecard - Measures That Drive Performance*. Harvard Business Review
- [45] Daniel L. (2007). *Data Warehousing and OLAP-A Research-Oriented Bibliography*
- [46] Thomsen E. (1997). *OLAP Solutions: Building Multidimensional Information Systems*. John Wiley & Sons
- [47] Han J., Kamber M. (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann
- [48] Herschel G. (2008). *Magic Quadrant for Customer Data-Mining Applications*. Gartner Inc
- [49] Johanne K. (2015). *Business Intelligence for Business Users*. IBM
- [50] Andrew M. (2008). *Understanding Open Source and Free Software Licensing*. O'Reilly Media
- [51] Verts T. (2013). *Open Source Software*. World Book Online Reference Center
- [52] Gay J. (2012). *Respects Your Freedom certified products*. Free Software Foundation
- [53] Phipps S. (2015). *Handing On The Baton*. Open Source Initiative
- [54] Tang A. (2014). *Top Advantages and Disadvantages of Open Source Software*. BestHostingSearch
- [55] Brodtkin J. (2007). *Open Source Start-up Upgrades Data Integration Software*. NetworkWorld
- [56] Rao L. (2010). *Open Source Software Company Talend Raises \$34M, Acquires Sopera*. Techcrunch

- [57] David M. (2011). *Pentaho 4 Unites Enterprise Business Intelligence and Data Integration*. Ventana Research
- [58] Grimes S. (2005). *Open-Source BI Startup Pentaho Makes Debut*. InformationWeek
- [59] Marchetto R. (2008). *Talend Open Studio vs Pentaho Kettle, a Comparison*
- [60] Swenson E. (2002). *Reports Made Easy with JasperReports*. JavaWorld
- [61] Urlocker Z. (2012). *JasperSoft BI Suite 2.0*. InfoWorld
- [62] Heffelfinger D. (2007). *JasperReports Tutorial/Getting Started Guide*. Ensode
- [63] Hague N., Tatchell J. (2008). *BIRT: A Field Guide to Reporting*. Addison-Wesley Professional
- [64] Bappoo P. (2010). *BIRT for Beginners*. Lulu
- [65] Weathersby J., Bondur T. (2008). *Integrating and Extending BIRT*. Addison-Wesley Professional
- [66] Rosenbaum S. (2010). *Open Source Reporting Review - BIRT, Jaspersoft, Pentaho*. Innovent Solutions
- [67] Smith J., Perry N., Meijer E. (2002). *Mondrian for .NET*. Drdobbs
- [68] Nolan C. (2008). *Manipulate and Query OLAP Data Using ADOMD and Multidimensional Expressions*. Microsoft
- [69] Microsoft (2001). *XML for Analysis Specification*. MSDN
- [70] Oracle (2011). *Introduction to the OLAP Java API*. OLAP Java API Developer's Guide
- [71] Oracle (2007). *Java JDBC API*. Java Documentation
- [72] Cho X. (2012). *Pivot4J - New UI Independent JPivot Replacement*. Pivot4J
- [73] Stoughton N. (2005). *Update on Standards*. USENIX
- [74] Bergsten H. (2013). *JavaServer Pages (3rd ed.)*. O'Reilly Media
- [75] Witten I., Frank E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition*. Morgan Kaufmann, São Francisco
- [76] Holmes G., Donkin A. (1994). *Weka: A Machine Learning Workbench*. Proc Second Australia and New Zealand Conference on Intelligent Information Systems
- [77] Deutsch G. (2010). *RapidMiner from Rapid-I at CeBIT 2010*. Data Mining Blog
- [78] Hamilton N. (2008). *The A-Z of Programming Languages: YACC*. Computerworld
- [79] Surani M. (2015). *Data Mining Tools (R, Weka, RapidMiner, Orange)*. Slideshare
- [80] Corcoran E. (2009). *The Big Deal: Tibco*. Forbes
- [81] Baker L. (2014). *Vista Equity to Buy Tibco Software for \$4.3 Billion*. Reuters
- [82] Staff T. (2013). *TIBCO Releases B2B Portal Platform*. Electronic
- [83] Actuate (2014). *BIRT iHub F-Type*. Sourceforge
- [84] Held B., Erb H. (2006). *Advanced Controlling with Excel. Corporate management with OLAP and PALO*, Franzis Poing
- [85] Müller S. (2010). *Business Intelligence in Sales on the Basis of Open Source Solutions*. Marketing- and Vertriebs
- [86] Sheina M. (2010). *Pentaho BI Suite Enterprise Edition*. Ovum
- [87] Moody G. (2010). *How Do You Make a Pentaho?*. Computerworld
- [88] Grimes S. (2005). *Open-Source BI Startup Pentaho Makes Debut*. InformationWeek
- [89] SpagoBI (2012). *Business Intelligence with SpagoBI*. SpagoBI Competency Center
- [90] Lacy M. (2010). *SpagoBI – Plataforma BI Livre e Aberta*. Revista Espírito Livre
- [91] Schroder C. (2012). *5 Best Linux Business Intelligence Suites*. Small Business Computing

- [92] Golfarelli M. (2008). *Open Source BI Platforms: a Functional and Architectural Comparison*. DEIS, University of Bologna
- [93] Thomsen C., Pedersen B. (2005). *A Survey of Open Source Tools for Business Intelligence*. DaWaK
- [94] Evelson B., Hammond J. (2010). *Open Source Business Intelligence (BI)*. Business Process Professionals
- [95] Anderson R. (2012). *Intro to ASP.NET MVC 3 (VB)*. Microsoft ASP
- [96] FitzMacken T. (2014). *ASP.NET 5 Overview*. Microsoft ASP.NET
- [97] Glenn K., Pope S. (1998). *A Cookbook for Using the Model–View–Controller User Interface Paradigm in Smalltalk-80*. The JOT (SIGS Publications)
- [98] Rayfield A., James T. (2001). *Web-Application Development Using the Model/View/Controller Design Pattern*. IEEE Enterprise Distributed Object Computing Conference
- [99] Herman D. (2013). *Effective JavaScript*. Addison-Wesley
- [100] Haverbeke M. (2011). *Eloquent JavaScript*. No Starch Press
- [101] Marcotte E. (2010). *Responsive Web Design*. A List Apart

## ANEXO I – SISTEMAS DE GESTÃO DE INFORMAÇÃO NAS ORGANIZAÇÕES

A necessidade de gerir a informação advém do facto de que "as organizações têm de encarar a incerteza e os eventos aleatórios provenientes internamente e externamente e contudo oferecer um sistema conceptual claro, operacional e bem organizado para os utilizadores" (Daft e Lengel, 1984).

Quanto mais genérico e organizado é o sistema de informação, interpretado como um conjunto de recursos humanos e técnicos, dados e processos, interligados entre si, com o objectivo de fornecer informação relevante para a gestão das tarefas da organização onde está embutido, e quanto melhor mostrar a organização em funcionamento, mais robusta será a organização, no sentido em que o SI vai agir sob a forma de análise da empresa e dos seus sistemas contextuais. O SI funcionará como um catalisador de mudança objectiva na estrutura da organização, revelando novos desafios e requerendo a utilização de metodologias aliadas às TI (Tecnologias de Informação), pois estas formam um potencial de desenvolvimento para as organizações.

As TI incentivam a evolução, trazem inovações e atraem novos investimentos. Ao mesmo tempo, conduzem a um aumento da eficácia e na redução dos custos, bem como na melhoria dos serviços prestados aos utilizadores e na sua qualidade. As TI são instrumentos essenciais na concepção de SI (Sistemas de Informação) integrados e organizados. Actualmente, os decisores de negócio têm que estar conscientes de que planeamento estratégico dos SI é crítico de criação de valor para a empresa.

Resumidamente, para os SI são priorizadas as necessidades de informação e sua inclusão nos negócios, através de uma análise da empresa e do seu contexto, bem como na análise da estratégia geral da organização. Os dados por si só não significam nada, é preciso transformá-los em informação, mas podemos ir mais além e tornar esta em conhecimento.

### Sistemas de Informação para a Gestão

Foi nos anos 70 que surgiram no mercado os primeiros sistemas interligados e automatizados de gestão [1], denominados de *Management Information Systems* (MIS [2]). Estes tinham como objectivo a integração da informação oriunda de vários processos de negócio das empresas. No entanto, nesta primeira fase, tornaram-se patentes muitas limitações das infra-estruturas existentes, assim como das linguagens de programação, sistemas operativos, *hardware*, entre outros. Todas estas tecnologias eram bastante arcaicas, quando comparadas às existentes nos dias que correm. A evolução tecnológica fez que estas dificuldades iniciais fossem ultrapassadas. Com o aparecimento dos PC (*Personal Computers*), houve um aumento, em todos os departamentos das empresas, da utilização de ferramentas informáticas. Depois das primeiras fases de informatização, as médias e grandes organizações passaram, desde o início do século, a estar focadas em projectos que pretendem melhorar a sua eficácia operacional e estratégica, através de:

- Sistemas operacionais integrados (ERP [3][4]) que permitiram às empresas otimizar a sua parte transaccional, otimizando os processos, transformando a informação e integrando áreas que antes operavam isoladamente;

- Implementação de projectos funcionais, como os *Customers Relationship Management* (CRM [5][6]) para a gestão das relações com os clientes e os *Supply Chain Management* (SCM [7]) para a gestão da cadeia de distribuição.

Os investimentos no tipo de sistemas anteriores foram, praticamente, melhorias feitas aos processos de *back-office* das empresas, desde a área das compras, passando pelas áreas de gestão de stocks e gestão de recursos humanos até à área de facturação.

### **Sistemas Inteligentes e o seu Valor Acrescentado**

Apesar das diferenças de dimensão de empresas do mesmo tipo (pequena, média e grande dimensão), no que toca à sua eficiência elas têm-se nivelado cada vez mais [7], pois já não chega às organizações serem competitivas transaccionalmente, elas têm que ser eficientes na dimensão corporativa. A estratégia passa no acesso, na partilha e análise da informação e na transformação da mesma em conhecimento útil para a tomada de decisão por parte dos gestores. Esta estratégia é uma visão integrada que tem como objecto os processos organizacionais de uma empresa, baseada, essencialmente, no seu ciclo de gestão: análise e gestão, orçamentação e planeamento, previsão, estabilização económica, *reporting* e análise da *performance*, e por último, gestão de resultados e compensações. Os sistemas inteligentes, incluindo as ferramentas de BI, têm três responsabilidades fundamentais:

- Acesso oportuno à informação;
- Capacidade de análise;
- Capacidade de *reporting*.

### **Processos Operacionais e Processos Estratégicos**

A eficiência operacional tem sido a grande preocupação dos gestores [8]. O número de projectos foi aumentando no sentido da automatização e integração dos dados transaccionais, pois o principal objectivo foi otimizar os processos de negócio no que toca ao tempo gasto, aos custos, riscos e qualidade final da informação gerada [9]. No que toca aos processos operacionais, devem-se destacar as áreas da logística, produção, comercial, financeira e recursos humanos, pois são estas áreas que tem a maior parte dos processos e rotinas diárias de consumo, transformação e disponibilização de informação, sustentando assim as actividades críticas das empresas. Os processos estratégicos responsabilizam-se por um grupo de actividades vitais para a sustentabilidade da empresa, como por exemplo: análise estratégica, planificação, estabelecer objectivos, desenvolvimento, controlo e monitorização, contabilidade e consolidação da informação. A Figura 1 mostra a grande diferença entre estes dois tipos de processos: do lado esquerdo, os processos operacionais preocupam-se com a realização eficiente das actividades; do lado direito, os processos estratégicos, preocupam-se com as actividades que devem ser feitas.

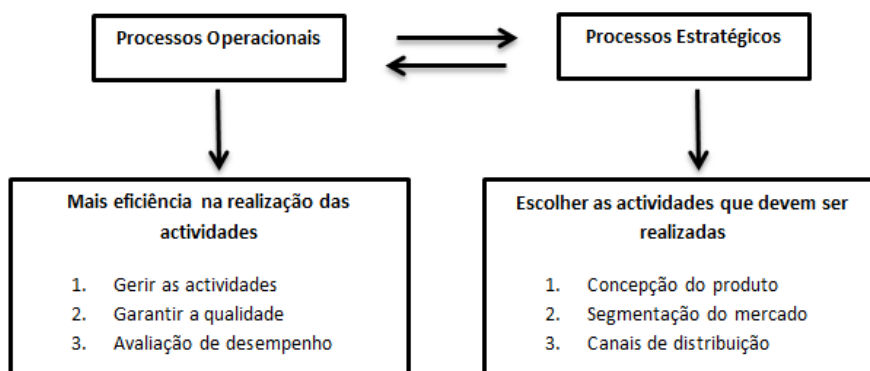


Figura 1 - Diferença entre os processos operacionais e os processos estratégicos

Os processos operacionais beneficiam dos esforços, modernizações e integrações que foram ocorrendo ao longo das últimas décadas. Já os processos estratégicos apenas tiveram mais foco e atenção, por parte dos gestores, nos últimos anos, porque só recentemente é que houve o aperfeiçoamento necessário para as ferramentas de apoio à tomada de decisão marcarem a diferença no mercado.

### Planear, Orçamentar e Prever

Na gestão das empresas actuais, a primeira *business pain* [8] (que se pode designar de “dor da gestão organizacional”) é dar resposta ao seguinte desafio: após se definir uma estratégia organizacional, como se faz todo o processo de planeamento de forma a fazer com que as operações sejam instrumentos úteis para a gestão da empresa?

A Figura 2 mostra o tradicional ciclo de planeamento de uma organização.

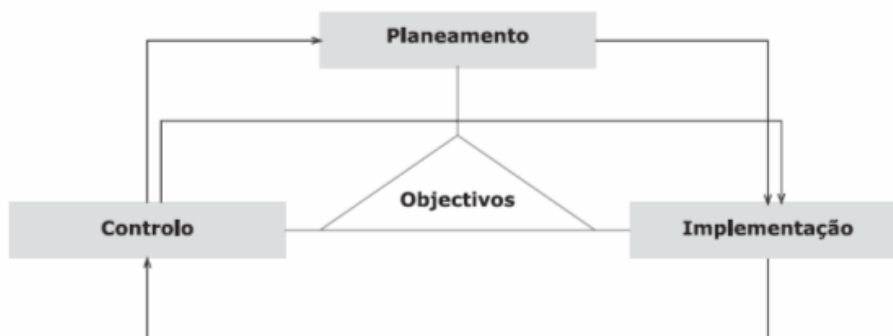


Figura 2 – Ciclo de planeamento (fonte: <http://agrogestao.com/pagina.asp?ID=65>)

Muito resumidamente, o que está em cima da mesa é: o processo, os objectivos e o suporte (ou seja, a tecnologia) da função de planeamento [9]. Em muitas empresas, o planeamento ainda continua a ser um processo moroso, centralizado e bastante inflexível. A separação, muito comum, entre o planeamento e a orçamentação faz com que os departamentos operacionais actuem de forma cada vez mais isolada, causando um desinteresse e desânimo por parte da organização em executar esta tarefa. O planeamento requer uma etapa de aquisição de dados, respectiva normalização e *reportings* finais em formatos bem conhecidos: orçamentos (os chamados *budgets*), resultados conseguidos (lucros e perdas), balanço e distribuição de *cash-flows*. No mundo dos negócios actuais, o que é desejável é a previsão dinâmica, ou seja, a existência de actualizações das previsões iniciais dependendo dos novos

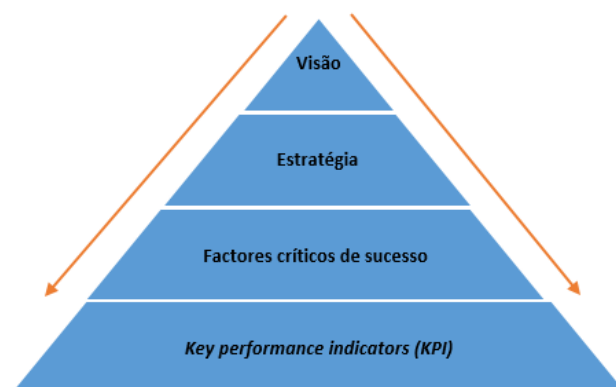
*inputs* de dados. As ferramentas e tecnologias tradicionais continuam a ser as mais utilizadas, ou seja, as folhas de cálculo. No entanto, estes ficheiros de Excel criam *bottlenecks* ao nível da criação, partilha, aquisição e consolidação dos dados, fazendo com que os processos se prolonguem por semanas ou meses, havendo o agravamento da interpretação diferente de cada analista e gestor de negócio, que pode fazer com que o *ping-pong* destas folhas de cálculo prolongue ainda mais o processo. Para lá dos problemas identificados anteriormente, existem outros, tais como as mudanças de contexto frequentes, erros nas fórmulas e *macros* de cálculo, ligações incorrectas entre folhas de cálculo diferentes ou (e provavelmente, o mais grave problema) alterações indevidas nos dados. Estes problemas fazem com que exista a necessidade de encontrar e corrigir os erros. As empresas que utilizam as soluções tradicionais descritas anteriormente têm dificuldade em efectuar o planeamento e controlo em tempo real. Os colaboradores passam maior parte do tempo a recolher e a organizar dados em vez de se preocuparem com a análise e preparação da tomada de decisão. Os objectivos para um sistema organizado tecnologia-processo são os seguintes [10]:

- Planeamento rápido e flexível;
- Trabalho de equipa e interajuda, em tempo real, de todos os intervenientes da organização;
- Capacidade de previsão a curto-médio prazo, sem necessidade de reformulações do processo no seu todo;
- Fiabilidade dos dados orçamentados para utilização por parte dos gestores internos.

Os objectivos anteriores podem ser respondidos por tecnologias de planeamento, controlo e monitorização, ou seja, as soluções de *Business Intelligence* são candidatas para a gestão dos processos de negócio das organizações [11].

### **Implementar a Estratégia, Gerir a Performance e Analisar a Informação**

Como se propaga pela empresa a informação sobre os objectivos, indicadores e metas e como se consegue gerir, em tempo real, o desempenho? Esta é considerada a segunda *business pain* [8]. É importante perceber, em primeiro lugar, a forma das organizações gerirem as suas actividades. Depois de ser definida a estratégia e definido o planeamento, a organização precisa de divulgar as acções esperadas dos diversos departamentos operacionais e dos colaboradores a nível individual. Resumidamente, o que se está a falar é da definição da visão, estratégia corporativa e dos *Key Performance Indicators* (KPI's). A Figura 3 mostra a pirâmide estratégica implementada nas organizações [12]:



**Figura 3 – Pirâmide estratégica: da visão aos indicadores**

Os indicadores devem ser inicialmente bem definidos e ao longo do tempo devem ser atentamente monitorizados. Alguns aspectos que devem ser levados em consideração:

- Comunicação e avaliação dos indicadores a partir das métricas das actividades de negócio;
- Controlo dos indicadores em curtos intervalos de tempo, de modo a que o ciclo planeamento-execução-controlo-análise seja rápido;
- Reforço da produtividade de cada responsável pelas suas áreas de responsabilidade, de modo a melhorar as acções operacionais.

Este controlo necessita de um modelo de gestão desempenho, e a sua execução através de ferramentas tecnológicas adequadas. A maior dificuldade passa pela aquisição, transformação e acessibilidade da informação. Muitos dos indicadores não são apenas números simples adquiridos de um único sistema fonte, até pelo contrário, é frequente que a informação seja composta e agrupada a partir de vários sistemas. A elevada quantidade e diversidade de factos e dimensões fazem com que a tarefa de análise dos dados seja complexa e as organizações acabem por demorar bastante tempo a tomar decisões. Neste momento, a questão mais pertinente a colocar é: Como se conseguem, saber, em tempo útil, as principais tendências de negócio, de modo a que as organizações possam tirar partido a partir destas conclusões e decisões evolutivas e correctivas? Esta questão torna-se ainda mais pertinente quando a organização tem centenas, milhares ou milhões de dados sobre as suas actividades, ou seja, quanto maior for a quantidade e complexidade, maior é a dificuldade em adquirir e analisar a informação.

Em suma, devem ser identificados os principais factos e dimensões de negócio, de modo a que a informação seja analisada, e após a análise desta a organização estará em condições de agrupar a informação em quadros únicos de métricas e relatórios. A Figura 4 mostra uma forma de análise de informação para responder a uma questão de negócio. A partir do cruzamento de dados, é possível verificar quem foram os vendedores das mais diversas categorias de produtos, num trimestre específico.

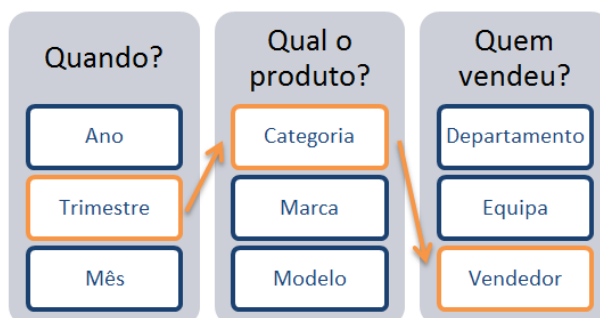


Figura 4 – Visão de negócio dos gestores

## Referências

- [1] Enticknap N. (1998). *Computing's Golden Jubilee*. Resurrection (The Computer Conservation Society)
- [2] Laudon C., Laudon P. (2009). *Management Information Systems: Managing the Digital Firm*. Prentice Hall/CourseSmart
- [3] Rouse M. (2015). *ERP (Enterprise Resource Planning)*. TechTarget

- [4] King. W. (2005), *Ensuring ERP Implementation Success*. Information Systems Management
- [5] Yun Z., Joseph W., David C. (2003). *Customer Relationship Management (CRM) in Business-to-Business (B2B) E-commerce*. Emerald 11
- [6] Bolton N. (1998). *A Dynamic Model of the Duration of the Customer's Relationship with a Continuous Service Provider: The Role of Satisfaction*. Marketing Science
- [7] Bartsch F. (2013). *Supply Chain Management (SCM)*. BB Handel
- [8] Jesus J., Rocha L., Viana R. (2001). *Avaliação de Pequenas e Médias Empresas e Gestão de Risco*. Faculdade de Engenharia da Universidade do Porto
- [9] Coker F. (2014). *Understanding the Vital Signs of Your Business*. Ambient Light Publishing. pp. 41–42
- [10] O'Brien A. (2003). *Introduction to Information Systems: Essentials for the e-Business Enterprise*. McGraw-Hill, Boston
- [11] Beynon-Davies P. (2009). *Business Information Systems*
- [12] Alter S. (2006). *The Work System Method: Connecting People, Processes, and IT for Business Results*. Works System Press

## ANEXO II – BASES DE DADOS E DATA WAREHOUSING

### Base de Dados – O Conceito

As bases de dados [1][2][3] não são mais do que um conjunto de dados organizados sistematicamente. Apesar de não ser obrigatório, a maioria das bases de dados estão guardadas em formato digital. Existem várias formas de modelar uma base de dados, os mais conhecidos são os modelos relacionais e os dimensionais (muito utilizados em *data warehouses*). A *Structured Query Language* (SQL [4]) é a linguagem utilizada para a consulta de dados de bases de dados em formato digital. Esta foi definida pelo padrão do American National Standards Institute (ANSI), no entanto existem várias versões, e algumas delas variam consoante o fabricante do *software* de gestão de bases de dados. A maioria das versões existentes cumpre os padrões base definidos pelo ANSI, utilizando as palavras-chave definidas por este instituto (p.e: SELECT, UPDATE, DELETE, INSERT e WHERE). A Tabela 1 **Error! Reference source not found.** mostra uma tabela fictícia que guarda dados de clientes.

Tabela 1 – Tabela de Clientes

COD_CLIENTE	NOME_CLIENTE	MORADA	LOCALIDADE
1	Hernâni Rafael	Rua das Flores, nº 26	Viseu
2	Tiago Silva	Rua da Escola, nº 27	Penalva do Castelo

Utilizando a linguagem SQL, poderíamos consultar o nome e morada dos clientes, que tivessem a localidade de Penalva do Castelo, da seguinte forma:

```
SELECT C.NOME_CLIENTE, C.MORADA
FROM CLIENTES C
WHERE C.LOCALIDADE = 'PENALVA DO CASTELO'
```

Para a manipulação e consulta de uma base de dados é necessário um Sistema de Gestão de Bases de Dados (SGBD [3]). Este sistema tem como objectivo gerir os ficheiros das bases de dados, e caso sejam uma aplicação do tipo servidor (p.e: Microsoft SQL Server ou Oracle) devem aceitar os pedidos das aplicações cliente de modo a gerir os dados existentes nestas bases de dados. As aplicações cliente não manipulam os dados directamente, estas enviam pedidos ao SGBD, e este é o responsável pela manipulação e aquisição dos dados. A designação de “base de dados” incluiu a base de dados em si e o *software* SGBD. No entanto, é importante distinguir estes conceitos, podemos dizer que a base de dados é o livro e o SGBD é o bibliotecário, o livro tem a história e a o bibliotecário é responsável por arrumar os livros e tratar dos mesmos. Os SGBD não são apenas responsáveis pela gestão e consulta dos dados das bases de dados, eles são também responsáveis por processar as transacções, recuperar dados, controlar os acessos, evitar a corrupção dos ficheiros, entre outras funcionalidades. Outra vantagem do tipo da arquitectura cliente-servidor é a redução drástica do consumo de largura de banda na rede onde operam pois todas as acções são feitas num servidor dedicado com o SGBD, e apenas são devolvidas às aplicações cliente a informação relevante que foi pedida por estas.

## Relações entre Tabelas

Os dados podem estar repartidos por várias tabelas e estas podem ser consultadas ao mesmo tempo através da linguagem SQL. Para se fazer esta operação é necessário que as tabelas tenham pelo menos um campo comum que as “ligue” [4]. A Tabela 2 e a Tabela 3 mostram uma tabela de clientes e outra tabela de moradas:

Tabela 2 – Tabela de Clientes (2)

COD_CLIENTE	NOME_CLIENTE
1	Hernâni Rafael
2	Tiago Silva

Tabela 3 – Tabela de Moradas

COD_MORADA	COD_CLIENTE	MORADA	LOCALIDADE
1	1	Rua das Flores, nº 26	Viseu
2	2	Rua da Escola, nº 27	Penalva do Castelo
3	2	Rua Santiago, nº 31	Sintra

Através da linguagem SQL poderíamos realizar uma consulta dos nomes e moradas dos clientes com localidade em Viseu, de uma forma idêntica à consulta anterior:

```
SELECT CL.NOME, MO.MORADA, MO.LOCALIDADE
FROM CLIENTES CL JOIN MORADAS MO ON CL.COD_CLIENTE = MO.COD_CLIENTE
WHERE MO.LOCALIDADE = 'VISEU'
```

No último exemplo, recorreu-se a uma relação entre duas tabelas através de um campo comum às duas tabelas (*cod\_cliente*), esta operação é denominada como junção (*join*) [4]. É através desta lógica de junções que assenta o modelo relacional. Quanto mais normalizada for uma base de dados, mais rápidas serão as actualizações dos dados nas tabelas [5][6], pois as redundâncias são eliminadas e as tabelas têm menos dados, logo a procura dos registos para actualizar é mais rápida [7]. No entanto para consultas complexas, como é o caso dos sistemas de OLAP [8][9], o processo de consulta de dados é mais lento, o modelo dimensional resolve este problema através da desnormalização dos dados.

## Data Warehouse – O Conceito

O conceito de *data warehouse* (DW [10][11]) tem associado alguns termos importantes, e para se perceber melhor o conceito de DW é necessário abordar estes termos. Um dos termos são os “dados”, num DW estes são constituídos por factos discretos que agrupam valores com natureza descritiva, quantitativa e qualitativa que são cruciais para o negócio a analisar. Outro termo é a “informação”, esta pode ser designada como um grupo de dados organizados que têm uma importância directa com o negócio, ou seja, a informação soma dados ao conhecimento já adquirido e com relevância para o desenvolvimento dos processos. O “conhecimento” é o terceiro conceito, e este é muito importante

para o desenvolvimento e concepção de qualquer tipo de projectos que envolva um DW. O conhecimento é a percepção e a interpretação que se possui em relação a um negócio, quanto maior for o conhecimento que as pessoas têm em relação a um negócio maior é a probabilidade de haver boas decisões a favor do mesmo.

Na literatura coexistem várias definições para o conceito de *data warehouse*, estas são oriundas de vários autores que se dedicaram muito a esta área, são exemplo disso mesmo o Bill Inmon [10] e Ralph Kimball [11]. A definição que mais consenso reúne entre os vários autores e especialistas na área é que um DW é um grupo de dados que apresentam as seguintes características [11]:

- **Dados integrados:** um DW deve apresentar-se como uma fonte de informação diversificada sobre e para o negócio que trata, ou seja, com um DW os utilizadores não deverão sentir a necessidade de consultar dados em diferentes fontes, todas as consultas importantes para o negócio deverão ser feitas unicamente ao DW;
- **Dados organizados por temas:** os dados devem ser agrupados por temas de acordo com os requisitos e necessidades do negócio a tratar;
- **Dados variam ao longo do tempo:** um DW deve guardar o histórico da informação ao longo do tempo de maneira a que os gestores consigam acompanhar e entender a evolução ao longo do tempo;
- **Dados não voláteis:** a informação presente num DW é estável. Quando os dados são inseridos no DW estes permanecem no mesmo não sendo apagados (excepto correcções a inserções incorrectas). Como os dados não são apagados, o *data warehouse* mantém o histórico da informação;
- **Dados acessíveis:** um dos objectivos principais de um DW (e do modelo dimensional) é permitir que os utilizadores acedam à informação facilmente e rapidamente.

Um *data warehouse* deve ser criado para dar resposta às necessidades dos utilizadores de um negócio. No que toca à aquisição de dados e análise da informação ele deve ser o mais eficiente possível de modo a que depois tenha matéria-prima fiável para a transformação desta em informação útil e relevante.

Um *data warehouse* é uma fonte de informação que agrega os dados históricos de natureza transaccional e operacional de um negócio, este armazenamento dos dados deve facilitar as tarefas de análise e de *reporting*.

Os dois motivos principais para a criação de um DW são [12]:

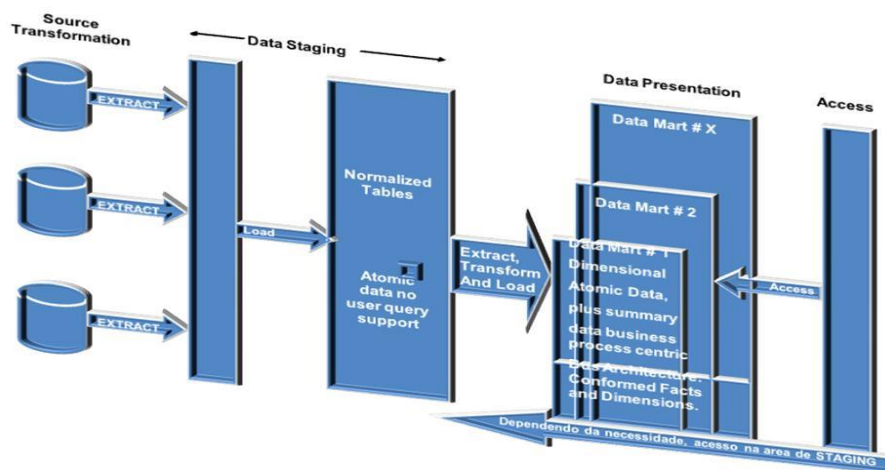
- Necessidade de integrar dados que se encontram espalhados por diferentes fontes de dados de modo a que se consiga fazer uma análise global dos mesmos;
- Necessidade de separar dados que são utilizados nos processos dos dia a dia (dados transaccionais) dos dados que são utilizados para análise e *reporting* e que posteriormente são utilizados para a tomada de decisões por parte dos gestores de negócio.

## Arquitectura de um Data Warehouse

O modelo dimensional é o tipo de modelação mais utilizada na construção de *data warehouses*, a correcta implementação deste modelo é um passo importante no ciclo de vida de um DW. O primeiro passo na construção de um DW é definir o modelo de dados. Após a criação deste modelo, os dados de uma ou mais fontes serão carregados no mesmo, o processo que “transporta” os dados é conhecido pelo nome de ETL. Qualquer que seja a estrutura física de um DW, para a mesma estar completa deve apresentar cinco funções principais [10]:

- **Recepção:** repositórios no DW têm como principal responsabilidade a recepção de dados das fontes origem. As fontes de dados origem podem apresentar-se sob a forma de tecnologias diferentes e a aquisição dos dados deve ser feita com uma frequência temporal bem definida. É ainda nesta etapa que é feita a limpeza os dados de modo a garantir a qualidade e relevância dos mesmos para posteriores análises;
- **Integração:** repositórios com este fim agregam dados de diversas fontes, por exemplo imaginemos que um produto num sistema origem tem o código  $x$  e noutra sistema o mesmo produto o produto tem o código  $y$ , um repositório com a função de integração tem como objectivo fazer a correspondência entre ambos. Neste passo é comum haver outra limpeza de dados, principalmente pelos casos deste género em que é necessário haver correspondência de dados iguais mas que chegam ao DW de uma forma “diferente”;
- **Distribuição:** estes repositórios têm como objectivo transferir dados para outros repositórios que entregam dados para análises de processos e questões de negócio directas. Os dados podem permanecer bastante tempo num repositório do tipo de distribuição;
- **Entrega:** os repositórios com a função de entrega têm como principal objectivo entregar os dados de acordo com um âmbito bem definido sobre um ou mais negócios a analisar, é comum que estes repositórios entreguem dados a *data marts*, cubos multidimensionais, *dashboards*, *reports*, etc.;
- **Acesso:** estes repositórios têm como objectivo fornecer dados directamente aos gestores de negócios, na maioria das vezes este tipo de repositórios são os mesmos que fazem a entrega de dados.

Existem várias arquitecturas de *data warehousing*, desde a arquitectura de Ralph Kimball até à arquitectura de Bill Inmon, não serão apresentadas todas neste documento, a arquitectura escolhida para análise é a arquitectura híbrida pois junta várias ideias de vários autores, a Figura 1 mostra a organização de um DW neste tipo de arquitectura [11]:



**Figura 1 – Organização de um DW segundo a arquitectura híbrida (fonte: <http://litolima.com/2010/01/07/modelagem-dimENSIONAL-conceitos-basicos/>)**

Nesta arquitectura o DW encontra-se dividido em duas zonas e existem dois momentos diferentes em que ocorre o processo de ETL. Uma das zonas é a zona de tratamento de dados (*data staging*) e a outra é a área de apresentação. A área de apresentação é onde se situam as tabelas com os dados que são acedidos pelas aplicações e gestores de negócio, estas últimas duas entidades só acedem aos dados nesta zona do DW. A área de *data staging* é a área onde são inseridos dados temporários oriundos das fontes de dados origem, os dados podem ser carregados para esta área já tratados ou então podem ser tratados posteriormente. É normal que nesta área existam réplicas das tabelas normalizadas dos sistemas fontes, ou então tabelas normalizadas com as colunas que interessam analisar posteriormente. Os dados numa primeira fase são inseridos na área de *data staging* e posteriormente são inseridos na área de apresentação prontos a serem analisados, ou seja, existem dois processos de ETL e podem existir *n* processos de tratamento de dados até aos dados chegarem à área de apresentação.

As principais vantagens da arquitectura híbrida são [11][12]:

- *Divide to conquer* no que toca ao tratamento de dados pois podem ser feitas várias “passagens” sobre os dados de modo a que estes sejam inseridos correctamente na área de apresentação;
- Dados são acedidos unicamente numa zona do *data warehouse* evitando possíveis equívocos na apresentação de dados diferentes presentes em zonas diferentes;
- Abordagem híbrida permite criar tabelas (nas duas zonas do DW) e processos de ETL orientadas ao assunto, abordagem que mistura *top-down* com *bottom-up*;
- Permite a coexistência de equipas diferentes no processo de desenvolvimento e carregamento de dados. Por exemplo uma equipa pode estar responsável pelo ETL dos dados até à zona de *data staging* enquanto que outra pode estar responsável pelo ETL até à zona de apresentação de dados.

As principais desvantagens da arquitectura híbrida são [11][12]:

- Processo de ETL faz-se mais do que uma vez podendo haver um esforço desnecessário em certas situações;
- Existência de tabelas normalizadas no DW;

- Processo mais moroso por não haver apenas a área de apresentação.

### Esquema em Estrela (*Star Schema*) e em Floco de Neve (*Snow-Flake*)

O esquema em estrela (*star schema*) [11][13] é, muito provavelmente, o esquema mais vezes usado no desenvolvimento de um *data warehouse*. Este esquema baseia-se na ligação entre várias tabelas do tipo dimensão a uma tabela de factos, sendo comum a existência de várias tabelas de factos num DW. O esquema lógico é bastante fácil de entender, as tabelas dimensão têm as características dos processos (e.g.: dimensão geográfica, dimensão tempo, dimensão produto, etc.), já as tabelas de factos guardam os factos do processo de negócio a analisar, as tabelas de factos ligam-se às tabelas dimensão através de chaves estrangeiras em relações do tipo *n-1*. Este esquema destacou-se ao longo dos anos por ser bastante simples de entender e por ter as seguintes vantagens [13]:

- A informação crucial a analisar (factos) está presente numa única tabela de fatos, evitando assim possíveis redundâncias de dados;
- Cada dimensão apenas precisa de uma chave primária para ligação com a tabela de factos (embora possam ter chaves múltiplas para o tratamento das alterações em dimensão lenta);
- Pequeno número de ligações entre tabelas;
- Manutenção reduzida.

A Figura 2 apresenta um exemplo de esquema em estrela:

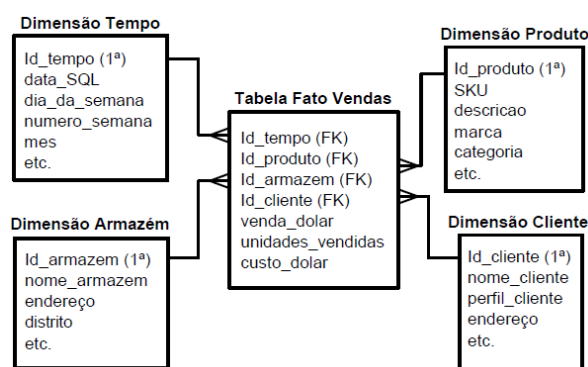
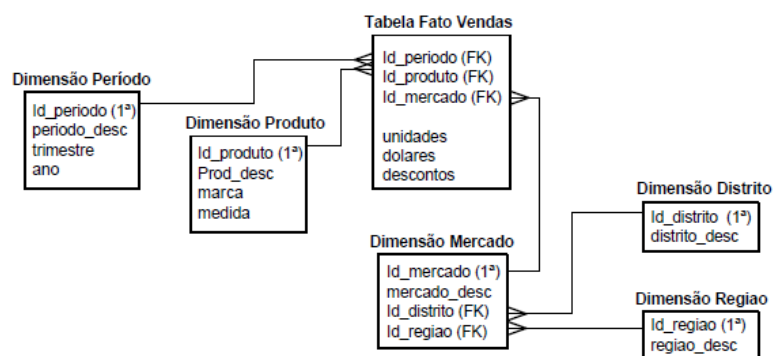


Figura 2 – Exemplo de esquema em estrela (fonte: <http://cacau-indicou.blogspot.pt/2012/02/data-warehouse-dw.html>)

Outro esquema, utilizado com alguma frequência, é o esquema em floco de neve (*snow-flake schema*) [11][12][13]. Este apresenta-se como uma extensão ao modelo em estrela. No modelo em floco de neve, cada “ponta” da estrela do modelo em estrela torna-se o centro de outras dimensões. Isto quer dizer que poderão existir dimensões interligadas entre si, no entanto só uma das tabelas deste conjunto é que se liga à tabela de factos, fazendo as ligações entre as dimensões passa a existir normalização de dados e consequentemente o acesso à informação torna-se mais lento devido à necessidade de fazer junções. Este modelo deve ser utilizado quando as dimensões começarem a ficar muito grandes e com muitos dados repetidos ao longo das linhas de registo, e em certos casos, se justifique a utilização de várias dimensões em vez de apenas uma e não se pretende ter dados redundantes em dimensões diferentes. Este tipo de esquema poderá apresentar-se como mais acessível na inserção de dados nas dimensões, pois as tabelas encontram-se normalizadas e é natural que se pareçam mais com as tabelas “originais” das fontes de dados transaccionais. No entanto, esta acessibilidade poderá fazer com que os

processos de ETL se tornem mais demorados e conseqüentemente exista a necessidade de mais manutenção. A Figura 3 apresenta um esquema em floco de neve:



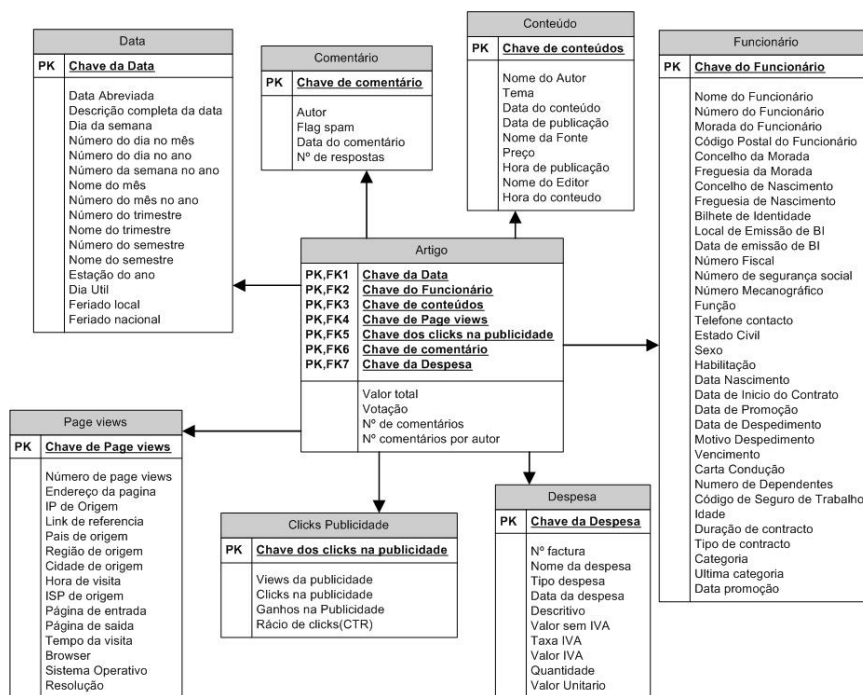
**Figura 3 - Exemplo de esquema em floco de neve**  
(fonte: <http://cacau-indicou.blogspot.pt/2012/02/data-warehouse-dw.html>)

A Tabela 4 apresenta uma análise comparativa entre estes dois esquemas [13]:

Tabela 4 – Análise comparativa entre o esquema em estrela e o esquema em floco de neve		
	Modelo em estrela	Modelo em floco de neve
<b>Tabelas dimensão</b>	Não normalizadas	Normalizadas
<b>Tamanho físico</b>	Maior (dados repetidos nas tabelas do tipo dimensão)	Menor que no modelo em estrela pois os dados encontram-se normalizados
<b>Velocidade das consultas</b>	Rápida	Menos rápida que o modelo em estrela por causa da necessidade de realizar junções

### Tabelas de Factos

As tabelas de factos [10][11], como o próprio nome indica, guardam os factos relevantes para analisar. É comum serem tabelas com grande volume de dados pois mantêm o histórico dos dados. Estas tabelas têm um conjunto de colunas que descrevem os factos. Por norma, estas colunas contêm dados numéricos que podem ser agregados (através de médias, somatórios, etc.). Além deste tipo de colunas é comum haverem colunas com chaves estrangeiras para as dimensões. As tabelas de factos são tabelas muito normalizadas, a chave primária é constituída por todas (ou parte) as chaves estrangeiras de ligação às dimensões e as restantes colunas são dados numéricos que “medem” os factos de negócio. Resumindo, estas tabelas representam transacções, acontecimentos, ou outras actividades, que são utilizadas para medir o funcionamento e os resultados dos processos de negócio. A Figura 4 mostra um exemplo de tabela de factos.



**Figura 4 – Exemplo de um esquema em estrela com uma tabela de factos “Artigo”**  
 (fonte: <http://ruisalgueirosousa.wordpress.com/2011/03/10/modelo-dimencional-para-data-warehousing-2/>)

Como se pode constatar pela figura anterior, existe uma tabela de factos “Artigo”, nesta existem sete chaves estrangeiras para as dimensões, sendo que estas chaves compõem uma chave composta que forma a chave primária, e os seus atributos são: o valor total, a votação, o número de comentários e o número de comentários por autor. O objectivo essencial desta tabela de factos é analisar, ao longo do tempo, as votações que diversos autores fazem a artigos.

## Tabelas de Dimensões

As tabelas das dimensões [10][11][12] contêm dados descritivos de entidades directamente relacionadas com os factos que se pretendem analisar. Na maioria das vezes estas tabelas, para lá da sua chave primária, contêm dados textuais, dimensões como “clientes”, “produtos”, “vendedores” e “localização geográfica” são bastantes comuns e todas elas contêm dados textuais que depois são ligados com os factos das tabelas de factos e assim fornecem a natureza descritiva aos dados numéricos. As dimensões para lá de dados descritivos podem conter dados numéricos (p.e: numa dimensão produto o preço do produto pode ser um atributo da dimensão), no entanto estes dados não costumam ser agregados como os dados numéricos das tabelas de factos. Estas tabelas têm, na maioria das vezes, menos registos que as tabelas de factos, não sendo de admirar existirem tabelas de factos com biliões de registos, por outro lado as tabelas de dimensões costumam ter no máximo milhares de registos.

Uma questão importante a abordar no que toca às tabelas dimensionais são as alterações em dimensão lenta, em inglês SCD (*Slowly Changing Dimensions*). As SCD são alterações feitas aos dados das dimensões numa grande espaço de tempo, daí o nome: alterações em dimensão lenta. Imaginemos o caso de um cliente que se registou no sistema como sendo solteiro sem dependentes, ao fim de quatro anos o mesmo cliente casou-se e teve dois filhos, esta alteração pode ser registada na tabela dimensional dos clientes. O facto do cliente ser agora casado e com filhos pode mudar os seus hábitos de consumo, é então importante que a ficha de cliente seja alterada de acordo com as novas

informações. Ao contrário dos sistemas transaccionais, onde podia ser feito um simples *update* ao registo do cliente, as tabelas dimensionais podem, tal como as tabelas de factos, guardar o histórico de informação. A possível mudança de hábitos de consumo pode ser relevante do ponto de vista histórico e um simples *update* da informação do cliente pode estragar o histórico e evolução de consumo do cliente. É então importante recorrer a técnicas específicas para tratar as SCD. Existem diversos tipos de SCD [12][13], os tipos 0, 1, 2, e 3 são os tipos mais usados, a escolha do tipo a utilizar deve ser ponderado de acordo com as necessidades e limitações do projecto e negócio a tratar.

### Tipo 0

Este tipo é o tipo mais fácil de entender, essencialmente porque nada acontece, é um método considerado passivo pois os dados não são alterados. Este tipo é o menos utilizado no que toca ao tratamento das SCD.

### Tipo 1

O tipo 1 é o mesmo que os métodos de actualização dos sistemas transaccionais, ou seja, os dados antigos são substituídos pelos novos, no entanto, o histórico dos dados é perdido. Este tipo é mais utilizado para correcções em dados incorrectos, ou, em dados que não são relevantes em guardar histórico, por exemplo, a Tabela 5 representa uma dimensão fornecedor com um registo que apenas tem o fornecedor “Colheitas São José, Ltd”.

Tabela 5 – Exemplo da dimensão fornecedor com um registo inicial

<b>Id</b>	<b>Cod_Fornecedor</b>	<b>Nome_Fornecedor</b>	<b>Distrito_Fornecedor</b>
1	001	Colheitas São José, Ltd	Viseu

Caso o fornecedor mude de nome poderá não ser necessário, do ponto de vista do negócio a analisar, guardar histórico dessa alteração, já uma alteração no distrito do mesmo poderá justificar-se. Uma SCD do tipo 1 apenas faz o *update* dos dados antigos com os novos, então a solução passaria por substituir o nome antigo pelo novo como está representado na Tabela 6:

Tabela 6 – Registo alterado com o novo nome do fornecedor

<b>Id</b>	<b>Cod_Fornecedor</b>	<b>Nome_Fornecedor</b>	<b>Distrito_Fornecedor</b>
1	001	Colheitas São José e Silva, Ltd	Viseu

### Tipo 2

Nas SCD do tipo 2 para cada alteração que é feita é inserido um novo registo, sendo que este novo registo tem uma chave nova. Com este tipo é mantido o histórico dos dados ilimitadamente pois para cada alteração é inserido um novo registo. Imaginemos agora que o fornecedor anterior muda o distrito para “Leiria”, com a alteração do tipo 2 a tabela dimensional ficaria como está demonstrado na Tabela 7.

Tabela 7 – Exemplo de SCD do tipo 2

<b>Id</b>	<b>Cod_Fornece dor</b>	<b>Nome_Fornecedo r</b>	<b>Distrito_Fornecedo r</b>	<b>Data_Inici o</b>	<b>Data_Fi m</b>	<b>Registo_Actu al</b>
-----------	----------------------------	-----------------------------	---------------------------------	-------------------------	----------------------	----------------------------

1	001	Colheitas São José e Silva, Ltd	Viseu	01-01-2015	04-05-2015	N
2	001	Colheitas São José e Silva, Ltd	Leiria	05-05-2015	31-12-9999	S

No tipo 2 é normal utilizarem-se datas de início e fim de actuação do registo, ou seja, colocar “datas de validade” dos registos, e para ajudar na pesquisa dos registos mais actuais costuma-se utilizar uma *flag* que indica se o registo é o mais actual, a existência destas datas e da *flag* estão representadas na tabela anterior.

### Tipo 3

O tipo 3 defende a criação de colunas adicionais que permitem guardar as sucessivas alterações que são feitas aos registos, imaginemos a alteração que foi dada como exemplo nas SCD do tipo 2 mas agora aplicada com o tipo 3, a Tabela 8 mostra como seria feita esta alteração

Tabela 8 – Exemplo de SCD do tipo 3

Id	Cod_Fornecedor	Nome_Fornecedor	Distrito_1	Distrito_2	Data_Alteracao_2	Distrito_3	Data_Alteracao_3
1	001	Colheitas São José e Silva, Ltd	Viseu	Leiria	05-05-2015		

Com este método, as alterações ficam registadas no mesmo registo mas em colunas diferentes, é comum a utilização de datas de alteração de modo a guardar o histórico temporal das alterações. Este método tem a grande limitação no que toca ao número de alterações que podem ser feitas a um registo, este número já não é ilimitado como no tipo 2.

### Chaves Substitutas

Num DW, por norma, as chaves primárias das tabelas são diferentes das chaves utilizadas nos sistemas origem. A política utilizada em *data warehousing* é o uso de chaves substitutas, em inglês SK (*Surrogate Keys* [10][12]). A chave substituta é essencialmente um código que identifica o registo no DW, é por isso a chave primária da tabela. A chave do sistema fonte pode ou não ser guardada, dependendo do significado que a mesma tem para a análise de dados no DW. Revendo o exemplo representado na **Error! Reference source not found.**, podemos constatar que o “Id” é a chave substituta da chave original que é o “Cod\_Fornecedor”. As chaves substitutas são as chaves estrangeiras das tabelas de factos. Quando uma tabela origem de uma dimensão possui uma chave composta para a identificação única da linha na tabela, usasse uma SK única para representar a chave primária composta. As principais vantagens destas chaves são [10]:

- **Imutabilidade:** as aplicações não perdem a referência ao registo da base de dados, uma vez que o identificador nunca muda;
- **Desempenho:** estas chaves costumam ter um tipo de dados compacto, otimizando assim as junções em comparação com as chaves compostas.

## Conceito e Importância dos Metadados

O conceito de metadados é importante de analisar, quando se fala em *data warehousing*. Metadados são, no sentido literal da palavra, “dados sobre os dados”, e representam o significado e características dos dados em si. Nas organizações, a existência de metadados é importante para a padronização, fiabilidade e coerência dos dados existentes nos sistemas de informação das mesmas. Numa base de dados relacional, a implementação e salvaguarda de metadados é feita, na maioria dos casos, através de tabelas descritivas. Estas guardam atributos que identificam as tabelas e os registos e têm outras colunas com as descrições os dados propriamente ditos. No âmbito de um DW, Ralph Kimball defende a existência de vários tipos de metadados [11]:

- **Metadados de negócio:** guardam dados sobre o negócio em si, ou seja, são registados dados sobre as áreas de negócio, métricas de negócio, estruturação dos dados, etc.;
- **Metadados dos processos:** registam a forma como os dados são extraídos, transformados e carregados no *data warehouse*;
- **Metadados técnicos:** são metadados puramente técnicos que guardam informação como os formatos dos dados, formatos dos ficheiros que são inseridos directamente no DW, estrutura das tabelas, etc.;
- **Metadados aplicativos:** guardam a maneira como se acedem e se utilizam os dados no DW. Este tipo de metadados pode também guardar quem e quando alguém acedeu aos dados.

A importância dos metadados é fundamental para as organizações nos seguintes aspectos [11]:

- Localização e significado dos dados;
- Monitorização e controlo do processo de extração, transformação e carregamento de dados;
- Validação da coerência e formato dos dados.

É essencial que exista um repositório para os metadados. Este repositório poderá estar dentro do DW em questão, num sistema isolado, ou em casos extremos, haver um único repositório de dados na organização que guarda os metadados de vários sistemas (até mesmo dos sistemas transaccionais). A criação do repositório de dados deverá ser feito através de várias iterações, assim o desenvolvimento torna-se mais fácil e, ao longo do tempo, a qualidade dos metadados pode melhorar.

## Etapas para a Construção de um Data Warehouse

O início de um projecto de *data warehouse* começa sempre com uma ou mais necessidades de negócio existentes numa organização [10][11]. As mais comuns são a gestão, análise e apresentação de dados respeitantes a processos importantes. Em alguns casos, a simples necessidade de *reporting* poderá justificar a existência de um DW. A partir do momento da especificação inicial poderá começar-se a desenvolver o *data warehouse*. No entanto, um projecto deste tipo poderá necessitar de várias iterações para estabilizar e poderão haver mudanças no negócio ao longo do tempo de desenvolvimento e manutenção. Isto tudo faz com que o ciclo de vida de um DW possa ser considerável. Por norma, é no início do projecto que se começam a levantar as primeiras dificuldades

de implementação e aprendizagem. No entanto, é também neste momento que os utilizadores começam a aperceber-se dos benefícios e vantagens de um DW.

### **Análise do Negócio e das Necessidades**

Um projecto de *data warehousing* é constituído por vários passos que não têm que ser obrigatoriamente abordados sequencialmente, e alguns deles, devido a questões como a dimensão do projecto e as regras de negócio envolvidas, podem ser mesmo ignorados. A primeira etapa passa por elaborar um *business case*, este deverá responder a questões como [10][11][13]:

- Quais são as necessidades dos utilizadores?
- Como é que o projecto vai contribuir para a resolução das necessidades?
- Quais serão os custos directos e indirectos do projecto?
- Como é que actualmente se faz a aquisição dos dados?
- Qual o nível de fiabilidade dos dados?
- Quais são os riscos que se poderão levantar?

Todas as organizações devem responder a estas questões antes do início de um projecto de *data warehousing*. A todas as questões é importante responder, mas a consciencialização dos custos do projecto é muito importante. Se os custos forem mal calculados, o projecto pode falhar.

A avaliação dos riscos é deveras importante, pois é através deste que a organização pode resolver possíveis problemas durante o desenvolvimento e manutenção do projecto. Esta avaliação deve tentar responder às seguintes questões [14]:

- A(s) tecnologia(s) que se vão utilizar têm provas dadas no mercado?
- A equipa de desenvolvimento tem o *know-how* necessário para implementar o projecto?
- Qual o tempo em que se obtém um ROI aceitável?

As respostas às perguntas anteriores são muito importantes, sendo através destas que se pode chegar à conclusão que o projecto deve ou não ser empreendido de acordo com as necessidades, custos e riscos. Um projecto que faça sentido, do ponto de vista das necessidades e da solução a que se propõe, pode não ser exequível pelos custos que virá a ter ou então os riscos do projecto são tantos que a probabilidade de insucesso pode fazer com que o projecto não se venha a concretizar.

A segunda fase para o desenvolvimento de um projecto de *data warehousing* é a avaliação das infra-estruturas da organização, desde os aspectos técnicos aos aspectos não técnicos. Os aspectos não técnicos passam pelos processos da organização, sendo exemplo disso as metodologias de desenvolvimento: os aspectos técnicos são, por exemplo, o *software* e *hardware* necessário para a implementação do projecto. Nesta fase levantam-se mais perguntas que devem ser respondidas pela organização que quer implementar um projecto de DW e de BI, são exemplo [14][15]:

- Qual o *hardware* e o *software* necessário e qual o custo dos mesmos?
- É preciso contratar mais pessoas para fazer a manutenção do *hardware*?
- É necessário comprar um SGBD e existe capacidade de manutenção do mesmo?

O correcto levantamento das necessidades de *hardware* e *software* é importante, mas tão ou mais importante é analisar a capacidade de resposta que a equipa de desenvolvimento terá, a necessidade de formação também deve ser considerada. A escolha do SGBD deve ser justificada, os factores que contribuem para a escolha vão desde a facilidade de utilização até ao desempenho, o aumento do espaço físico que os dados vão ocupar também deve ser levado em conta. A resposta à perguntas anteriores são de teor técnico. Existem perguntas de teor não-técnico que devem ser também respondidas, são exemplo [15]:

- Quem são os responsáveis pela criação e manutenção das fontes de dados e dos modelos lógicos?
- Será criado um repositório de metadados? Existe necessidade para tal?
- Existe documentação das fontes de dados? É política da organização fazer e manter esta manutenção?
- Será feita a limpeza dos dados com vista a qualidade dos mesmos? Quem serão os responsáveis? Existe *software* para *data quality*?

Os modelos lógicos dos dados são muito importantes num DW, pois são estes que “criam” as regras de negócio e a normalização dos dados. Nesta etapa devem ser considerados com atenção, assim como os metadados necessários para o novo sistema.

A terceira fase é o planeamento do projecto. Neste, todos os intervenientes devem ser envolvidos por dois motivos: colaboração e envolvência. A colaboração entre os demais intervenientes é essencial para o sucesso de qualquer projecto, mas no caso da criação de um *data warehouse* é ainda mais importante, pois poderão estar envolvidas pessoas de áreas e com competências completamente diferentes, sendo também essencial a partilha dos conhecimentos existentes sobre o negócio a tratar. Esta fase de desenvolvimento de um DW é constituída por quatro sub-fases: levantamento dos requisitos, análise dos dados, prototipagem e análise do repositório de metadados (se for criado). A definição de requisitos é essencial em qualquer projecto, no âmbito de um projecto de *data warehousing* devem encontrar-se respostas para as seguintes questões [15]:

- Qual a informação que os utilizadores finais precisam?
- Quais as questões que actualmente não conseguem ser respondidas e para os quais existe a necessidade de resposta? Porque que não se consegue actualmente responder a estas perguntas?
- Como é que são obtidos os dados que os utilizadores precisam?
- Os dados nos sistemas fontes têm qualidade e encontram-se actualizados?
- Qual a granularidade que os dados precisam no DW e qual o tempo necessário para histórico de dados?
- Existirão *reports* urgentes que têm que ser tirados de um dia para o outro?
- Quantos utilizadores e com que frequência utilizaram as aplicações de BI para consultas os dados?

A análise de dados é outra etapa importante num projecto de *data warehousing*. Pode definir-se como sendo a análise detalhada dos sistemas fontes que identifica possíveis falhas na qualidade dos dados, de maneira a se identificarem quais as melhores fontes para o DW e conseqüentemente para as

aplicações de BI. Esta etapa costuma consumir muito tempo e esforços por parte da equipa de DW/BI. Aqui o processo iterativo pode não ser a melhor metodologia, pois o ciclo de vida deste processo pode estender-se bastante. Por norma, corrigem-se todos os dados importantes e apenas mais à frente se corrigem os dados menos importantes. Nesta etapa levantam-se as seguintes perguntas [10][11]:

- Os mesmos dados podem ser extraídos de fontes diferentes? Se sim qual a fonte mais estável (qualidade e actualização dos dados)?
- Quem são os responsáveis pelos dados de cada sistema fonte? Conseguem garantir a fiabilidade e qualidade dos dados?
- Quais são os dados cruciais a extrair? Estes dados têm a qualidade mínima e aceitável para a inserção no DW?

A prototipagem é outra etapa que deve ser considerada na implementação de um projecto de DW. Esta tem o objectivo de dar a conhecer as ideias gerais que a equipa tem para o DW. Um protótipo de DW tem na maioria das vezes referência à aplicação de BI. Neste mostram-se as ideias que existem para as *interfaces* da aplicação e os dados que serão mostrados (assim como a forma como serão apresentados: *reporting*, *dashboarding*, OLAP, etc.). O tempo de desenvolvimento dos protótipos deve ser o menor possível e estes devem ser os mais directos e esclarecedores possíveis. Outro aspecto importante é o detalhe dos protótipos. Este deve ser bem ponderado o nível de detalhe: protótipos muito detalhados podem fazer com que exista uma sensação de compromisso por parte dos clientes ao produto final. No caso do produto final ser bastante diferente dos protótipos (detalhados) pode surgir alguma contestação dos utilizadores. Por outro lado, um protótipo pouco detalhado pode ser pouco esclarecedor daquilo que se pretende fazer. Em resumo, deve então ser bem pensado o nível de detalhe que os protótipos terão.

A última etapa da fase de planeamento do projecto é a análise do repositório dos metadados. É nesta etapa que são identificados os metadados que deverão ser guardados. Já foi discutida a importância dos metadados neste documento, faltou referir que estes são utilizados tanto por utilizadores com cariz mais técnico, como por exemplo a(s) equipa(s) de desenvolvimento e manutenção do DW e das aplicações de BI, como pelos utilizadores finais. Os metadados ajudam os utilizadores finais a interpretar e a localizar os dados existentes num ambiente de BI, estes por exemplo ajudam os utilizadores a encontrarem os dados e as relações entre os mesmos dentro de um DW e até mesmo dentro de uma aplicação de *Business Intelligence*. As respostas às seguintes perguntas ajudam a fazer o repositório de metadados [11]:

- Quais os tipos de metadados necessários?
- Quem será o encarregado pela manutenção do repositório?
- Quem terá acesso, e de que forma, ao repositório?
- O repositório será utilizado apenas pelos utilizadores do DW ou será também utilizado pelas aplicações de *Business Intelligence*?

### **Desenho do *Data Warehouse***

Depois da análise de negócio e das necessidades, e de todas as etapas inerentes a esta fase, chega a hora de desenhar o DW propriamente dito. Aqui aparecem mais três etapas: desenhar a base de dados que forma o DW, desenvolver o processo de ETL e criar o repositório de dados.

No desenho da base de dados deverá ser ponderado o esquema a seguir (se o esquema em estrela, floco de neve ou outro) e a arquitectura do *data warehouse*. Independentemente do esquema ou arquitectura escolhidos, existem passos comuns que devem ser feitos. Exemplos:

- Construção das tabelas de factos para cada processo de negócio a analisar;
- Definição da granularidade dos dados, isto é, o nível de pormenor dos dados. Esta é uma questão essencial pois uma granularidade desajustada pode pôr em risco as análises a fazer ou a *performance* das mesmas;
- Identificar as dimensões dos dados;
- Ponderação dos atributos das dimensões no que toca às descrições necessárias para contextualizar a análise dos factos;
- Preparar, tanto as tabelas de factos como as dimensionais, no que toca à manutenção do histórico dos dados.

Para executar os passos anteriores, devem-se responder às seguintes questões:

- Qual o nível de agregação (tabela de factos) necessário para as análises futuras?
- Qual o ritmo esperado no que diz respeito ao crescimento físico da base de dados?
- Será necessário a criação e utilização de índices nas tabelas? Existirá a necessidade de particionamento das tabelas?
- Em que medida o SGBD escolhido influencia o desenho do DW?

Depois do DW estar desenhado é hora de se pensar na implementação do processo de ETL. O objectivo deste processo é extrair dados de diversas fontes (nos mais diversos formatos) que posteriormente são inseridos no *data warehouse* e serão a fonte das análises e *reports* das aplicações de BI. Em suma o processo de ETL é um grupo de processos que permite a extracção de dados de diversas fontes de informação, estes são transformados e carregados noutras fontes de informação (ficheiros, base de dados relacionais ou dimensionais, etc.).

Um processo de ETL [16][17], como o próprio nome indica, é constituído por três fases: extracção, transformação e carregamento, segue-se a descrição de cada uma destas etapas [17]:

- **Extracção:** aquisição de dados que se encontram numa fonte de informação, seja de que formato for (ficheiros excel, bases de dados relacionais, bases de dados orientadas ao documento, etc.). Nesta fase os dados a extrair podem ser todos ou apenas os dados que foram modificados desde a última vez que se extraiu informação;
- **Transformação:** consiste em “limpar” os dados de modo a que sejam inseridos com a maior qualidade possível nos sistemas destino. O processo de transformação poderá implicar a realização de várias operações como: mapeamento de valores, agregação de dados, transformação selectiva (apenas para certas categorias e subcategorias de dados), etc.;
- **Carregamento:** esta fase apenas insere registos no sistema destino, como por exemplo, o *data warehouse*. Tratamento das SCD do tipo 1 podem complicar o carregamento, visto que têm que ocorrer actualizações de dados.

Das três fases anteriores a mais desgastante, tanto a nível e tempo como de esforço, é a transformação de dados. Nesta fase por vezes é preciso fazer mapeamento complexos ou *fixing* de dados incorrectos. Depois de extrair, transformar e carregar os dados é ainda necessário e importante fazer diversos testes. A correcção atempada de erros detectados nesta fase facilita o desenvolvimento de todo o processo em si, correcções futuras ao *go live* do processo de ETL podem tornar-se muito demorosas e pode comprometer *timings* ao nível da gestão de projecto.

Dois aspectos importantes que se devem ter atenção quando se está a desenhar um processo de ETL são [16]:

- **Simplicidade e escalabilidade:** os requisitos podem ser alterados a qualquer altura é então importante que os processos de ETL sejam desenhados de forma a que consigam, no futuro, ser facilmente alterados e expandidos;
- **Desempenho:** é preciso ter a consciência que um processo pode demorar desde segundos a horas. É então necessário que em certos casos, principalmente para os processos mais morosos, a *performance* seja levada em conta, de maneira a que o processo demore e interfira o menos possível nas actividades dos utilizadores finais.

Depois do DW estar construído e com dados testados e fiáveis é tempo de colocar os mesmos a serem visualizados em aplicações e/ou *reports*. As aplicações que usam dados de um *data warehouse* não têm que ser, obrigatoriamente, de *Business Intelligence*, podem ser simples *reports* em formato *office* (p.e: word e excel). Na fase de construção de aplicações ou métodos que lêem dados do DW, devem-se encontrar respostas para as seguintes questões [15]:

- Deverá existir uma aplicação *web*?
- Devem-se ter em conta aspectos de segurança de dados e controlo de acesso?
- Qual a experiência de utilização dos utilizadores finais?
- Será preciso dar formação aos utilizadores?

Depois de todas as fases vistas anteriormente chega a hora do *Go Live* final, pode-se dar por finalizado o projecto de DW e BI (caso as aplicações e/ou *reports* sejam desta natureza) e começa a fase de manutenção.

## Referências

- [1] Ling L., Tamer M. (2009). *Encyclopedia of Database Systems*. Illus
- [2] Beynon P. (2004). *Database Systems. 3rd Edition*. Palgrave
- [3] Connolly T., Begg C. (2002) *Database Systems*.
- [4] Codd E.F. (1970). *A Relational Model of Data for Large Shared Data Banks*. Communications of the ACM 13
- [5] Date J. (2003). *An Introduction to Database Systems*. Addison Wesley
- [6] Gray J., Reuter A. (1992). *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers
- [7] Kroenke M., David A. (2007). *Database Concepts*: Prentice
- [8] Daniel L. (2007). *Data Warehousing and OLAP-A Research-Oriented Bibliography*
- [9] Thomsen E. (1997). *OLAP Solutions: Building Multidimensional Information Systems*. John Wiley & Sons

- [10] Inmon B. (1992). *Building the Data Warehouse*. Wiley
- [11] Kimball R. (1996). *The Data Warehouse Toolkit*. Wiley
- [12] Ganczarski J. (2009). *Data Warehouse Implementations: Critical Implementation Factors Study*. VDM Verlag
- [13] Date C. *An Introduction to Database Systems*. Addison Wesley
- [14] O'Brien A. (2003). *Introduction to Information Systems: Essentials for the e-Business Enterprise*. McGraw-Hill, Boston
- [15] Alter S. (2006). *The Work System Method: Connecting People, Processes, and IT for Business Results*. Works System Press
- [16] Oracle (2007). *Overview of Extraction, Transformation, and Loading*. Database Data Warehousing Guide
- [17] Polsku P. (2007). *ETL Process*. Datawarehouse4u

