



UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

WEBDATAMART
PARA
GESTÃO PORTUÁRIA

Paulo Miguel Silvério Santos
(Licenciado)

Dissertação para obtenção do grau de Mestre em:

Engenharia Informática e de Computadores

Orientador: DOUTOR ALBERTO MANUEL RODRIGUES DA SILVA

Júri

Presidente: DOUTOR ALBERTO MANUEL RODRIGUES DA SILVA
Vogais: DOUTOR MÁRIO JORGE COSTA GASPAR DA SILVA
DOUTOR PÁVEL PEREIRA CALADO

Novembro de 2005

Resumo

A necessidade de analisar a informação operacional produzida pelas organizações não é um problema recente, mas, tem vindo a agudizar-se significativamente. O *data warehousing* consiste na tecnologia encontrada que melhor se adequa para armazenamento e análise de dados históricos. O *business intelligence*, por outro lado, pretende ajudar os decisores no processo de tomada de decisão, nomeadamente através de ferramentas de OLAP e de ferramentas de *reporting*.

Nesta dissertação parte-se de um caso real, de um sistema transaccional denominado SITEP, para a área dos transportes marítimos, que permite registar todos os acontecimentos relevantes que se desenrolam nos portos numa base de dados relacional. Para além disso, o SITEP permite a produção de relatórios com fins estatísticos. Contudo, para além do facto de o tipo do modelo de dados utilizado pelo SITEP não ser o mais adequado para a realização de análises, o método de produção de relatórios revela-se pouco flexível.

O objectivo desta dissertação assenta na criação de uma base de dados multidimensional (adequada aos processos de análise) e no estudo de mecanismos que permitam flexibilizar o processo de tomada de decisão na gestão portuária, nomeadamente através do desenvolvimento de um conjunto de operações OLAP e de um método de produção dinâmica e interactiva de relatórios.

No processo de transformação de dados entre a base de dados operacional e a base de dados multidimensional (a residir nos Analysis Services) são utilizados os Data Transformation Services. São também discutidas diversas alternativas de armazenamento para os dados (MOLAP, ROLAP e HOLAP). Quanto ao conjunto de operações OLAP são disponibilizadas as operações mais recorrentes (*roll up*, *drill down*, *slicing* e *dicing*). Relativamente à produção de relatórios é utilizada a nova plataforma de *reporting* da Microsoft, os "Reporting Services".

Palavras chave: *Business Intelligence, Data Mart, Data Warehousing, ETL, OLAP, Reporting*

Abstract

The need to analyze the operational information produced by the organizations is not a recent problem, but, it has become more urgent. The data warehousing consists in the technology that better adjusts for storage and analysis of historical data. Business intelligence, on the other hand, intends to help the decisores in the decision making process, nominated through OLAP and reporting tools.

In this essay we start with a real case, of a transactional system called SITEP, for the area of the maritime transports, that allow to register all the relevant events that occur in the ports in a relational database. For moreover, SITEP allows the production of reports with statistical ends. However, for beyond the fact of the type of the model of data used by the SITEP not to be adjusted for the accomplishment of analyses, the method of production of reports shows little flexible.

The goal of this essay is based on the creation of a multidimensional database (adjusted to the analysis processes) and in the study of mechanisms that allow to become more flexible the decision making process in the port management, nominated through the development of a set of OLAP operations and a method of dynamic and interactive production of reports.

In the data transformation process between the operational database and the multidimensional database (stored in Analysis Services) are used the SQL Server Data Transformation Services. Also diverse alternatives of storage for the data are argued (MOLAP, ROLAP and HOLAP). Relatively to the set of OLAP operations are implemented the most recurrent operations (roll up, drill down, slicing and dicing). Relatively to the production of reports the new platform of reporting of the Microsoft, the "Reporting Services" is used.

Keywords: Business Intelligence, Data Mart, Data Warehousing, ETL, OLAP, Reporting

Agradecimentos

Num trabalho deste género é, de certo modo, normal surgirem algumas adversidades que, por vezes, nos levam a pensar que o fim é inatingível. Algumas vezes tive esse pensamento, mas com a ajuda de algumas pessoas, de quem não me vou esquecer jamais, tudo foi ultrapassado e penso que atingi o objectivo a que me propus.

Por isso, quero deixar expressos o meu reconhecimento e gratidão a todos aqueles que de alguma forma contribuíram para a sua realização, em particular:

Ao orientador do trabalho, Prof. Alberto Silva, pela sua reiterada disponibilidade, espírito de abertura, rigor científico, encorajamento e crítica constantes, que permitiram levar a cabo esta dissertação;

A todos com quem trabalhei, em particular ao Prof. João Matos, pelas facilidades que me concederam para que pudesse estar presente nas diversas reuniões de preparação da dissertação;

À Administração do Porto de Aveiro pela sua contribuição em termos técnicos e disponibilização de informação;

Por último, mas acima de tudo, à minha família e aos meus amigos pelo apoio incondicional durante todo o processo de realização desta dissertação, que sempre acreditaram que tudo tem um fim.

A todos o meu sincero “Obrigado”.

Lisboa, Novembro de 2005

Paulo Miguel Silvério Santos

Dedicatória

*Aos meus pais e
à minha irmã*

Conteúdo

1. INTRODUÇÃO
2. ENQUADRAMENTO TEÓRICO
3. CASO DE ESTUDO: GESTÃO PORTUÁRIA – DMPORT
4. ARMAZENAMENTO E TRANSFORMAÇÃO DE DADOS
5. OPERAÇÕES DE OLAP E DE REPORTING
6. AVALIAÇÃO E DISCUSSÃO DE RESULTADOS
7. CONCLUSÕES E TRABALHO EM ABERTO

Índice

Capítulo 1

Introdução.....	1
1.1. Enquadramento	1
1.2. Motivação	3
1.3. Caso de Estudo.....	5
1.4. Definição do Problema.....	6
1.5. Questões Orientadoras da Investigação.....	7
1.6. Organização da Dissertação.....	7

Capítulo 2

Enquadramento Teórico	9
2.1. Conceitos Gerais	9
2.1.1. <i>Data Warehousing</i>	9
2.1.2. <i>Data Mart</i>	11
2.1.3. Conceitos OLAP.....	12
2.1.4. Operações OLAP.....	15
2.2. Armazenamento Multidimensional.....	18
2.2.1. MOLAP (<i>Multidimensional Data Storage</i>)	19
2.2.2. ROLAP (<i>Relational Data Storage</i>)	19
2.2.3. HOLAP (<i>Hybrid Data Storage</i>)	19
2.2.4. Comparação entre MOLAP, ROLAP e HOLAP.....	20
2.3. Análise de Ferramentas de <i>Data Warehousing</i>	21
2.3.1. Microsoft SQL Server 2000 Analysis Services.....	21
2.4. Análise de Ferramentas de <i>Reporting</i>	24
2.4.1. Microsoft SQL Server 2000 Reporting Services.....	25

Capítulo 3

Caso de Estudo: Gestão Portuária – DMPORT.....	29
3.1. Introdução ao Caso de Estudo	29
3.2. Modelo de Dados e Funcionalidades Anteriores	32
3.3. Requisitos com Nova Configuração.....	35

Capítulo 4

Armazenamento e Transformação de Dados.....	37
4.1. Introdução.....	37
4.2. ETL com o Microsoft SQL Server 2000: DTS.....	38
4.3. DTS aplicado ao Caso de Estudo.....	41
4.3.1. O <i>package</i> “Leitura dos ficheiros dBase 5”	43
4.3.2. O <i>package</i> “Criação e actualização das tabelas de dimensões”	44
4.3.3. O <i>package</i> “Processamento dos mov. De navios, mov. De mercadorias, entidades e ocupação de cais”	47
4.3.4. O <i>package</i> “Processamento dos quadros para o Eurostat”	48
4.4. Dificuldades, Problemas e Soluções	49

Capítulo 5

Operações de OLAP e de <i>Reporting</i>	53
5.1. OLAP vs <i>Reporting</i>	53
5.2. Enterprise Blocks	55
5.3. Operações de OLAP no DMPORT	58
5.4. Operações de <i>Reporting</i> no DMPORT	65
5.4.1. <i>Reporting</i> com o Crystal Reports	65
5.4.2. <i>Reporting</i> com os Reporting Services	67

Capítulo 6

Avaliação e Discussão de Resultados	75
6.1. Transformação de Dados	75
6.2. Armazenamento de Dados	76
6.3. Operações OLAP	78
6.4. Operações de <i>Reporting</i>	78

Capítulo 7

Conclusões e Trabalho em Aberto	81
7.1. Conclusões do Trabalho	81
7.2. Trabalho em Aberto	83
7.3. Comentário Final	84

Apêndices

A.1 Lista de cubos construídos	1
A.1.1 Cubos com informação exigida pelo Eurostat	1
A.1.2 Cubos com registos de movimentos	4
A.1.3 Cubos com informação sobre indicadores	5
A.2 Modelo Multidimensional (base de dados dmpportDW)	7
A.3 Glossário	11
A.4 Referências	13

Lista de figuras

Capítulo 1

Figura 1.1 - Abordagem tradicional do processo de tomada de decisão (via informação operacional).....	4
Figura 1.2 - Abordagem actual do processo de tomada de decisão (via informação decisional).....	5

Capítulo 2

Figura 2.1 - Estrutura genérica de um cubo.....	12
Figura 2.2 - Esquema em estrela exemplificativo.....	13
Figura 2.3 – Interpretação visual do cubo.....	14
Figura 2.4 – Tabela de factos e tabelas de dimensões com os respectivos valores.....	15
Figura 2.5 – Interpretação visual para a operação de <i>slice</i>	17
Figura 2.6 – Interpretação visual da operação de <i>dice</i>	18
Figura 2.7 – Exemplo da edição de um cubo através da ferramenta <i>Cube Editor</i>	22
Figura 2.8 – Exemplo da pré-computação das agregações através do <i>Storage Design Wizard</i>	23
Figura 2.9 – Arquitectura dos Reporting Services.....	26

Capítulo 3

Figura 3.1 – Organização de um porto em terminais e cais (adaptado de [Planibase94]) .	30
Figura 3.2 – Tráfego de navios e de mercadorias (adaptado de [Planibase94])	30
Figura 3.3 – Interação entre as entidades envolvidas na gestão portuária (adaptado de [Planibase94])	31
Figura 3.4 – Quadro F1 – Mov. Navios por Tipo de Operação, Tipo de Navio e Classe de Deadweight	33
Figura 3.5 – Informação operacional de suporte às operações portuárias.....	34
Figura 3.6 – Cubo EuroF1 (Movimento de Navios por Tipo e Classe de <i>Deadweight</i>)	36

Capítulo 4

Figura 4.1 – Área de trabalho do DTS que possibilita a construção visual de um <i>package</i>	41
Figura 4.2 – Etapas do processo de transformação de dados.....	42
Figura 4.3 – <i>Package</i> referente à leitura dos ficheiros DBF.....	44
Figura 4.4 – <i>Package</i> referente à criação e actualização das tabelas de dimensões	46
Figura 4.5 – <i>Package</i> referente ao processamento dos movimentos de navios, movimento de mercadorias, entidades e ocupação de cais	47
Figura 4.6 – <i>Package</i> referente ao processamento dos quadros para o Eurostat	48
Figura 4.7 – Relação entre o cubo EuroFx e o cubo virtual EuroF1	49
Figura 4.8 – <i>Snapshot</i> da tabela de movimentos de navios	49

Capítulo 5

Figura 5.1 – Arquitectura do Enterprise Blocks [adaptado de [Eblocks-]].	56
Figura 5.2 – <i>Snapshot</i> da utilização do componente PivotTable	58

Figura 5.3 – Áreas de trabalho do componente PivotTable.....	59
Figura 5.4 – Interface gráfica disponibilizada na construção de análises OLAP	61
Figura 5.5 – Modelo de objectos do ADOMD [adaptado de [Microsoft-]].....	62
Figura 5.6 – Resultado final de uma análise através de uma tabela HTML.....	63
Figura 5.7 – Resultado final de uma análise através do <i>software</i> Enterprise Blocks.....	64
Figura 5.8 – Quadro EuroF1 – Movimento de Navios por Classe de <i>Deadweight</i> e Direcção segundo Ano.....	64
Figura 5.9 – Quadro EuroF1 – Movimento de Navios segundo Ano e Direcção por Tipo de Navio	65
Figura 5.10 – A solução RDC [adaptado de [Oliveira03]].....	66
Figura 5.11 – Diagrama de componentes dos Reporting Services.....	68
Figura 5.12 - <i>Snapshot</i> da utilização do Report Designer no Visual Studio .Net.....	69
Figura 5.13 – <i>Snapshot</i> da aplicação Report Manager	69
Figura 5.14 – <i>Snapshot</i> da interface gráfica para a criação de relatórios personalizados ..	70
Figura 5.15 – Diagrama de componentes relativo à construção de um relatório personalizado	71
Figura 5.16 – <i>Snapshot</i> da interface gráfica para a criação de relatórios personalizados (2)	71
Figura 5.17 – Visualização do relatório com o Report Viewer através do formato em tabela (matriz)	73
Figura 5.18 – Visualização do relatório sob a forma gráfica com o <i>Report Viewer</i>	74

Lista de tabelas

Capítulo 2

Tabela 2.1 – Unidades vendidas em todas as lojas de Portugal.....	16
Tabela 2.2 – Resultado da operação de <i>Roll Up</i> sobre a dimensão Tempo aplicada à Tabela 2.1.....	16
Tabela 2.3 – Resultado em tabela para a operação de <i>slice</i> aplicada à dimensão Loja=Faro	17
Tabela 2.4 – Resultado em tabela da operação de <i>dice</i> para a dimensão “Loja=Faro ou Loja=Porto”.....	18

Capítulo 5

Tabela 5.1 – Principais diferenças entre OLAP e <i>Reporting</i>	55
Tabela 5.2 – Tipos de serviços disponibilizados na arquitectura do Enterprise Blocks	57

Lista de siglas

API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
DBF	<i>DataBase File</i>
DTS	<i>Data Transformation Services</i>
ETL	<i>Extraction, Transformation, Loading</i>
FTP	<i>File Transfer Protocol</i>
GUI	<i>Graphical User Interface</i>
HOLAP	<i>Hybrid On-Line Analytical Processing</i>
IW	<i>Information Warehousing</i>
KPI	<i>Key Performance Indicators</i>
MDX	<i>Multidimensional Expression</i>
MS	<i>Microsoft</i>
OLAP	<i>OnLine Analytical Processing</i>
OLTP	<i>OnLine Transaction Processing</i>
OWC	<i>Office Web Components</i>
MOLAP	<i>Multidimensional On-Line Analytical Processing</i>
RDL	<i>Report Definition Language</i>
ROLAP	<i>Relational On-Line Analytical Processing</i>
SGBD	Sistema de Gestão de Base de Dados
SITEP	Sistema de Tratamento Estatístico Portuário
SQL	<i>Structured Query Language</i>
XML	<i>Extended Markup Language</i>

“O futuro não é um destino determinado pelo desenvolvimento da tecnologia, mas obra do Homem.”

Adam Schaff

“A ciência será sempre uma busca e jamais uma descoberta. É uma viagem, nunca uma chegada.”

Karl Popper

Capítulo 1.

Introdução

Sumário

Neste capítulo introdutório são apresentados os principais factores que motivaram esta dissertação. É discutido o problema e as respostas que a dissertação procura obter e são apresentadas as principais contribuições que esta investigação se propõe oferecer. Por fim, é descrita a organização da dissertação.

1.1. Enquadramento

A necessidade de analisar a informação diária operada pelas organizações não é um problema recente, mas, acima de tudo, tem vindo a agudizar-se [Jones98]. De acordo com o Gartner Group, em 2012, as grandes empresas terão de gerir trinta vezes mais informação do que a gerida actualmente [Gartner-]. Num mercado competitivo é necessário responder atempadamente a alterações ou tendências de mercado, satisfazer as necessidades dos clientes de forma célere e descobrir nichos de mercado onde a oferta ainda não seja significativamente expressiva.

No início dos anos 90 foi promovida a ideia da necessidade de implementar armazéns de informação, em inglês *information warehousing* (IW) [Biere03]. Em vez de transformar a informação existente em informação nova e útil, a ideia é deixá-la no mesmo local e acedê-la

a partir de ferramentas ou aplicações especializadas para o efeito. Surgiram na altura diversos tipos de tecnologias capazes de realizar análises sobre a informação existente. Este tipo de aplicações, responsáveis pela análise da informação, enquadra-se na área do *business intelligence*.

O conceito de *business intelligence* (BI) não é novo [Brandel01] [Lynn01], embora actualmente o conceito esteja enraizado no mundo das tecnologias de informação mais do que nunca. O BI é composto por uma vasto leque de aplicações e tecnologias que permitem a aquisição, armazenamento, análise e disponibilização de informação com o objectivo de ajudar os gestores e decisores a tomar melhores medidas na condução dos seus negócios [Greyner01]. As aplicações BI incluem as actividades dos sistemas de suporte à decisão, *query e reporting*, *online analytical processing*, análise estatística, *forecasting* e *data mining*. O *business intelligence* consiste na tecnologia que auxilia os gestores a tomar melhores decisões e de forma mais rápida de modo a superar as necessidades, citadas anteriormente, existentes num mercado competitivo.

Transversalmente ao *business intelligence* surgiu o conceito de *data warehousing* [Tan03]. O *data warehousing* tem um paradigma totalmente diferente do do *information warehousing* (IW). O *data warehousing* faz uma separação entre os dados transaccionais e os dados históricos, isto é, os dados que já não são necessários no desenrolar dos processos de negócio diários mas que servem de suporte ao processo de tomada de decisão, pois reflectem o passado da organização relativamente ao conjunto de opções e medidas adoptadas. O *data warehousing* será abordado com maior profundidade no capítulo 2 desta dissertação.

Dos vários tipos de aplicações *business intelligence* existem dois que merecem maior atenção da nossa parte, não só pela expressividade que têm vindo a adquirir nos últimos tempos, mas também porque se enquadram no âmbito desta dissertação: são eles o *online analytical processing* e o *reporting* [MicroStrategy-].

O *online analytical processing* (OLAP), foi criado por Codd em 1993 e tem por base o modelo de dados multidimensional [Agrawal97]. Segundo este modelo, que será descrito em maior detalhe no Capítulo 2, a informação encontra-se dividida em dois grupos distintos: os factos (valores mensuráveis do negócio, também designados por métricas) e as dimensões (perspectivas do negócio). Existe um conjunto de dimensões segundo as quais a informação de negócio pode ser analisada. Esta visão multidimensional assemelha-se ao modo como os gestores encaram a sua organização [Codd93].

O *reporting* é provavelmente o tipo de aplicações mais utilizado na área do *business intelligence*. Qualquer organização possui no mínimo uma ferramenta que possibilita a produção de relatórios quer textual quer graficamente, como por exemplo os relatórios de contas anuais, os relatórios de vendas ou os relatórios de despesas.

Para que este tipo de aplicações possa ter aplicabilidade é necessário existir um repositório onde a informação se encontre armazenada segundo o modelo multidimensional. Uma vez que os dados operados diariamente se encontram armazenados, na sua generalidade, em bases de dados relacionais, é necessário existir uma transformação e posterior passagem desses dados para bases de dados multidimensionais. Esta tarefa é designada ETL (*Extraction, Transformation, Loading*) [Boyno03][Inmon96][Trujillo03]. Os processos ETL são responsáveis pela extracção de informação a partir de fontes de dados operacionais heterogéneas, pela sua transformação (conversão, limpeza, normalização, etc.) e pelo seu carregamento em *data warehouses*. Estes processos serão abordados no Capítulo 4 da dissertação.

1.2. Motivação

Na grande maioria dos casos, as organizações utilizam nos seus sistemas de processamento transaccional o modelo relacional de dados. Embora outros modelos possam ser adoptados tal como o modelo orientado a objectos, o modelo relacional é na realidade o mais utilizado. Os sistemas de processamento transaccional (ou em inglês, *online transaction processing* – OLTP) são um tipo de sistemas especificamente desenhados para a captura diária de informação e rápidas actualizações. O modelo relacional é um tipo de modelo de dados adequado aos sistemas de processamento transaccional, uma vez que, segundo este modelo, a informação encontra-se normalizada [Codd70].

A crescente importância da necessidade de analisar a informação, de forma a tomar decisões mais coerentes, levou à concepção e desenvolvimento de sistemas de suporte à decisão, em inglês *decision support systems* (DSS) [Vaihov04]. Para Power [Power04] os DSS são uma classe de sistemas de informação que suportam o processo de tomada de decisão. Estes sistemas são interactivos e assistidos por computador com o propósito de ajudar os decisores utilizando tecnologias, informação, documentos, conhecimento e/ou modelos a completar o processo de tomada de decisão. As principais características de um DSS são: (1) é utilizado por gestores (ou decisores); (2) é utilizado para tomar decisões; (3) é utilizado para ajudar

peças, e não para as substituir; (4) é utilizado em decisões semi-estruturadas [Morton78] ou não estruturadas [Bonczek81]; (5) pode incorporar dados provenientes de uma base de dados de qualquer tipo; e (6) pode incorporar modelos.

Numa abordagem tradicional, tal como foi referido na secção anterior, os sistemas de suporte à decisão desenvolvidos acediam directamente à informação existente nas bases de dados dos sistemas transaccionais.

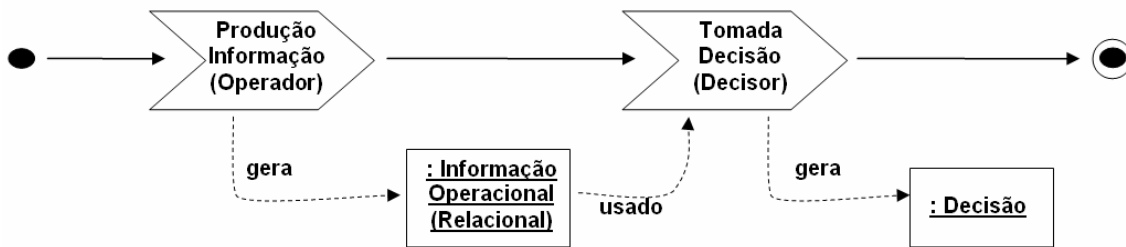


Figura 1.1 - Abordagem tradicional do processo de tomada de decisão (via informação operacional)

Segundo esta abordagem, ilustrada pela Fig.1.1 a informação produzida pelos operadores, no desenrolar dos seus processos de negócio diários, era armazenada nas bases de dados dos sistemas operacionais, na maioria dos casos segundo o modelo relacional. Posteriormente, essa mesma informação era utilizada no processo de tomada de decisão por decisores através de ferramentas especializadas para o efeito (DSS's). Embora esta abordagem fosse possível, não era a mais correcta uma vez que levantava um conjunto de problemas associados ao modelo de dados relacional. Cada tipo de modelo de dados é adequado ao tratamento de um problema em concreto. Os principais problemas enfrentados pela utilização de informação segundo o modelo relacional no processo de tomada de decisão são [Chaudhuri]:

- O modelo relacional é adequado aos sistemas transaccionais que exigem tempos de resposta muito reduzidos da ordem dos segundos, pelo que a realização de análises a grandes volumes de dados sobre esse modelo resulta num desempenho inaceitável.
- O suporte à decisão implica a consulta de informação histórica e as bases de dados operacionais contêm apenas informação actual.
- O suporte à decisão implica a consolidação de informação a partir de fontes heterogéneas, que podem conter dados com diferentes representações, o que implica a existência de um processo de transformação dos dados.

Houve então necessidade de criar um novo modelo de dados onde não se verificassem estes problemas. O modelo adoptado foi o modelo multidimensional, segundo o qual a informação

encontra-se descrita por um conjunto de dimensões e de métricas, ou seja, segundo diversas perspectivas [Agrawal97].

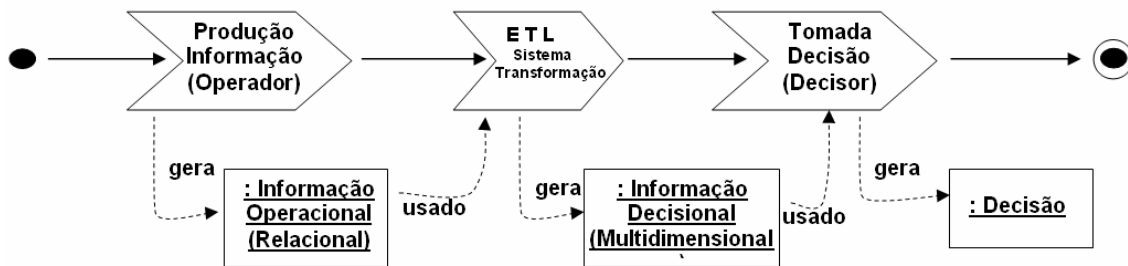


Figura 1.2 - Abordagem actual do processo de tomada de decisão (via informação decisional)

O processo de tomada de decisão passou a aceder à informação sobre o modelo multidimensional (ver Fig. 1.2). Esta nova abordagem permite resolver todos os problemas relacionados com a utilização do modelo relacional e é a utilizada hoje em dia pelas aplicações *online analytical processing* (OLAP).

A passagem entre os dois tipos de modelos de dados (modelo relacional e modelo multidimensional) obriga à existência de um processo intermédio. Este processo é responsável pela leitura dos dados a partir dos sistemas transaccionais, transformação dos mesmos e carregamento nos sistemas decisoinais, e designa-se por ETL (*Extraction, Transformation, Loading*), como já foi referido na secção anterior.

O processo de tomada de decisão é um processo aplicável às mais diversas áreas de negócio. A área dos transportes marítimos, especificamente a gestão portuária, que irá ser abordada nesta dissertação e brevemente introduzida na secção seguinte, é apenas um dos muitos cenários concretos da aplicabilidade dos sistemas de suporte à decisão.

Nesta dissertação partiu-se de um caso concreto, na área da gestão portuária, onde existe um sistema transaccional que utiliza o modelo relacional de dados e pretende-se transformar a informação para o modelo multidimensional de forma a melhorar o processo de tomada de decisão.

1.3. Caso de Estudo

A actividade portuária identifica-se por um conjunto de interacções entre diferentes tipos de interlocutores ou entidades. O universo em análise é composto pelas entidades e por um

conjunto de acções que se desencadeiam entre as diferentes entidades. Nos portos podem ocorrer diferentes tipos de tráfegos. Por exemplo, tráfego de navios comerciais, de navios de pesca, de mercadorias, de contentores, de passageiros.

O ponto de partida desta dissertação teve por base um produto informático desenvolvido pela empresa Planibase, designado de SITEP (Sistema de Tratamento Estatístico Portuário)¹ [Planibase94]. Este sistema permite que todas as acções ocorridas em cada porto sejam registadas regularmente numa base de dados relacional através dos seus utilizadores. Além disso, o SITEP permite a produção de um conjunto de relatórios com fins de análise estatística. O SITEP e a sua base de dados foram alvo da investigação levada a cabo na presente dissertação.

A partir da informação contida na base de dados do SITEP pretende-se construir um repositório de informação e desenvolver um conjunto de operações com capacidades analíticas que permitam, entre outros objectivos, maximizar a rentabilidade dos portos e proporcionar uma gestão portuária o mais adequada possível.

1.4. Definição do Problema

Partindo da base de dados do SITEP e assumindo que se pretende transformar a informação existente no modelo multidimensional somos confrontados com um primeiro problema: “Qual a técnica que se deverá utilizar para realizar o processo de transformação de dados entre as duas representações de dados?”. A discussão da técnica mais correcta a utilizar durante este processo é um dos focos da presente dissertação. Além disso, sabemos que a informação sobre o modelo multidimensional pode ser armazenada segundo diferentes formas (a saber, MOLAP, ROLAP e HOLAP)². Precisamos, por isso, de decidir qual dos tipos de armazenamento é o mais adequado para o caso de estudo.

Como já foi referido anteriormente, o SITEP permite registar todos os acontecimentos relevantes que se desenrolam nos portos, numa base de dados relacional. Embora o SITEP consiga produzir relatórios com fins estatísticos, os relatórios produzidos são “estáticos”, ou seja, a informação presente no relatório não depende do critério do utilizador mas sim de quem desenhou os relatórios. O método de produção de relatórios revela-se pouco flexível.

¹ No Capítulo 3 é feita uma apresentação mais detalhada do caso de estudo, nomeadamente do SITEP, em que se baseou esta dissertação

² Os tipos de armazenamento multidimensional serão abordados na Secção 2.2

Pretende-se por isso, estudar também mecanismos de flexibilizar o método de produção de relatórios.

Outro dos pontos fulcrais desta dissertação consiste no desenvolvimento de um conjunto de operações OLAP capazes de facilitar a análise sobre a informação portuária.

1.5. Questões Orientadoras da Investigação

De uma forma resumida as principais hipóteses colocadas no âmbito desta investigação são as seguintes:

- Qual é o modelo de dados mais adequado ao processo de tomada de decisão nesta área da gestão portuária? Qual a melhor técnica de armazenamento de informação para a realização do processo da tomada de decisão no contexto da gestão portuária?
- Qual a técnica que se deve utilizar para realizar o processo de transformação de dados entre os dois modelos de dados e repositórios de informação?
- Como disponibilizar um conjunto de operações capazes de auxiliar o processo de tomada de decisão? Como desenvolver um conjunto de operações OLAP de forma a flexibilizar os processos de análise? De que forma é que se pode aumentar a flexibilidade da produção de relatórios já existente no SITEP?

1.6. Organização da Dissertação

Esta dissertação encontra-se organizada em sete capítulos, que em seguida se apresentam de forma resumida.

No Capítulo 1 (“**Introdução**”) é definido o problema que se pretende abordar, fazendo referência ao caso de estudo utilizado, e por fim é apresentada a organização da dissertação.

O Capítulo 2 (“**Enquadramento Teórico**”) visa introduzir alguns conceitos gerais necessários à compreensão da investigação levada a cabo e apresentada na presente dissertação.

No Capítulo 3 (“**Caso de Estudo: Gestão Portuária – DMPort**”) é introduzido o caso de estudo utilizado nesta dissertação.

No Capítulo 4 (“**Armazenamento e Transformação de Dados**”) são abordados os tipos de armazenamento de informação multidimensional, ou seja, os tipos de armazenamento para o modelo de dados utilizado em *data warehousing* e pela tecnologia OLAP. Em seguida, é abordada a fase de transformação de dados entre bases de dados relacionais e bases de dados multidimensionais. Por fim, descreve-se pormenorizadamente o conjunto de *packages* desenvolvidos para a transformação dos dados relativos ao caso de estudo utilizado.

O Capítulo 5 (“**Operações de OLAP e de Reporting**”) trata dos dois tipos de aplicações de *business intelligence* utilizados na dissertação. Neste capítulo são descritas as operações de OLAP e de *reporting* implementadas no âmbito do caso de estudo utilizado.

No Capítulo 6 (“**Avaliação / Discussão de Resultados**”) são avaliados e discutidos os resultados obtidos nesta dissertação e propostos atingir no Capítulo 1.

No Capítulo 7 (“**Conclusões e Trabalho em Aberto**”) apresentam-se as principais conclusões que este trabalho identificou. Assim, são sumariadas e revistas as principais contribuições da investigação e são apresentadas as suas conclusões. No final são sugeridas futuras linhas de trabalho baseadas na investigação desenvolvida.

*“Todas as verdades são simples de compreender quando são descobertas;
a questão é descobri-las.”*

Galileu Galilei

Capítulo 2.

Enquadramento Teórico

Sumário

Este capítulo introduz os conceitos considerados relevantes à compreensão da investigação desenvolvida. Para além disso é realizada uma análise às ferramentas utilizadas durante a investigação.

Assim, na primeira secção são introduzidos conceitos de âmbito geral, nomeadamente os conceitos de *data warehousing*, *data mart*, conceitos OLAP e operações OLAP. Na segunda secção são abordados os diferentes tipos de armazenamento multidimensional e é feita uma breve comparação entre eles. Por último, na terceira e quartas secções são feitas breves análises às ferramentas de *data warehousing* e de *reporting*, respectivamente, utilizadas neste trabalho.

2.1. Conceitos Gerais

2.1.1. *Data Warehousing*

Não é necessário recuar muito no tempo para lembrar que certas decisões de executivos eram tomadas com base em intuições (“*feelings*”). Se existe uma chave para a sobrevivência

do negócio no novo milénio, ela passa certamente por ser capaz de analisar, planear e reagir a alterações de negócio de um modo mais rápido mas baseado em números e factos. Para atingir este objectivo, os gestores e analistas das empresas necessitam de informação acessível e adequada.

As tecnologias de informação revolucionaram a forma como as organizações operam a nível mundial. No entanto, embora muitas organizações hoje em dia disponham de computadores poderosos em todas as secretárias, e redes que interligam todas as partes do globo terrestre, muitos executivos e analistas não conseguem fazer chegar às suas mãos informação crítica e importante que já existe dentro da organização. A informação encontra-se espalhada por vários sistemas o que a torna extremamente difícil de a obter. Este fenómeno tem sido descrito pela “crise de acesso aos dados” (*the data access crisis*) [David02].

Hoje em dia, mais do que nunca, gestores executivos e analistas sabem que a informação armazenada em bases de dados departamentais necessita de ser unificada para que possam tomar decisões com a maior rapidez e eficiência possíveis. A informação precisa de estar disponível para leitura mesmo por utilizadores não técnicos para que estes também possam ajudar os seus superiores na tomada de decisões.

A par destas necessidades surgiu uma nova tecnologia de informação baseada no conhecimento, designada por “*data warehousing*”.

Um *data warehouse* pode ser definido com base em nove características que o diferenciam de um sistema operacional [David02]:

- **Disponibilidade:** o objectivo de um *data warehouse* é tornar a informação acessível ao utilizador.
- **Dados históricos:** a informação num *data warehouse* não é passível de ser alterada; a informação é inserida e não é mais modificada ou apagada.
- **Variável no tempo (*time-variant*):** toda a informação de um *data warehouse* tem uma componente temporal; o *data warehouse* contém informação que representa o estado de um negócio num período particular de tempo.
- **Separação:** o *data warehouse* está separado dos sistemas operacionais da organização; a informação armazenada é obtida a partir dos sistemas operacionais.
- **Integração:** a base de integração da informação espelha o modelo *standard* da organização.

- **Consistência:** a estrutura e conteúdo dos dados é bastante importante e apenas pode ser garantida através da utilização de metadados; estes metadados são independentes da origem dos dados e da altura de carregamento dos dados no *data warehouse*.
- **Orientada por assuntos:** a maior parte da informação é orientada por temas de negócio.
- **Iteração:** o *data warehouse* é uma implementação de um conceito; o *data warehouse* inicia-se pequeno e vai crescendo ao longo do tempo; o ponto de partida depende da área de interesse e a implementação é obtida através de um processo repetitivo.
- **Desempenho das agregações:** o sistema tem de dar resposta a diferentes níveis de agregação.

O *data warehouse* aparece como uma resposta tecnológica aos problemas e restrições existentes nos sistemas operacionais, uma resposta aos problemas de integração das aplicações operacionais e aos modelos de dados dos sistemas operacionais que não permitiam aos analistas e decisores das empresas, a criação de relatórios e análises adequadas, devido à informação não se encontrar armazenada de uma forma adequada para esses fins [David02].

2.1.2. *Data Mart*

Data warehouse e *data mart* são definidos e utilizados com fins distintos em diferentes sistemas de *data warehousing*. Não existe uma definição consensual à volta destes termos. No entanto, no âmbito desta dissertação assumimos as seguintes definições [Hoffer02]:

- *Data Warehouse* – conjunto de todos os dados de uma organização utilizado pela área de *business analysis queries*.
- *Data Mart* – conjunto de dados mais restrito de uma organização utilizado pela área de *business analysis queries* no âmbito de um departamento ou de um grupo de trabalho da organização.

Existem várias formas de estruturar a informação para utilização em *business analysis*, de entre as quais se refere:

- Estrutura *Non-Architected* – *data marts* independentes; esta é tipicamente a estrutura de um *data warehouse* desenvolvida sem um plano prévio.
- Estrutura de *Data Marts Dependentes* – *data warehouse* central com *data marts* dependentes; estrutura proposta por Bill Inmon [Inmon99].
- Estrutura de *Data Warehouse em Bus* – *data marts* ao longo do *data warehouse*; estrutura proposta por Ralph Kimball [Kimball98].

2.1.3. Conceitos OLAP

É importante distinguir os objectivos e capacidades de um *data warehouse* dos de um sistema OLAP (*OnLine Analytical Processing*). Os termos OLAP e *data warehouse* são complementares. Um *data warehouse* armazena e gere dados, enquanto que um sistema OLAP interroga e transforma os dados existentes num *data warehouse* em informação estratégica. Muitos conceitos são utilizados na modelação multidimensional e nos sistemas OLAP, designadamente:

- Cubo, cubo virtual, *star-schema*
- Tabela de factos e tabelas de dimensões
- Métricas
- Níveis, hierarquias e membros de uma dimensão

As funcionalidades de um sistema OLAP baseiam-se em análises dinâmicas de dados multidimensionais. Esses dados multidimensionais encontram-se armazenados em estruturas designadas de cubos. Diz-se portanto que um **cubo** é a estrutura básica de um *data warehouse*. Podemos então definir um **cubo** como sendo a unidade fundamental para armazenamento e visionamento da informação. A figura seguinte ilustra a estrutura genérica de um cubo.

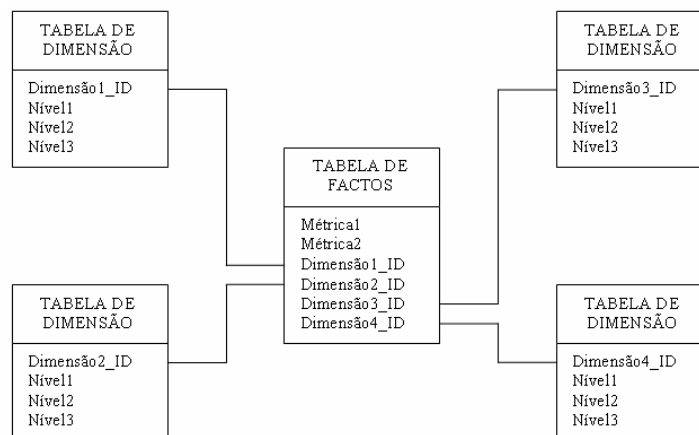


Figura 2.1 - Estrutura genérica de um cubo

Fazendo uma analogia aos sistemas transaccionais um cubo será o elemento equivalente a uma tabela relacional, ou seja, uma colecção de dados com atributos comuns. No entanto, um cubo não tem campos, mas sim dimensões que são definidas por quem constrói o *data warehouse*. Intuitivamente pensamos num cubo como tendo apenas três dimensões. Mas um

cubo pode ter menos ou mais de três dimensões. Um cubo é simplesmente uma metáfora para o armazenamento de dados.

Uma **dimensão** corresponde a uma perspectiva ou entidade que diga respeito à forma como se pretende guardar os dados. Para uma organização que se dedicasse à venda de certos produtos, poderíamos considerar como possíveis dimensões: a localização das suas lojas, o tipo dos produtos vendidos, ou o instante de tempo em que os produtos seriam vendidos. Associadas às dimensões estão as **métricas** que são os valores mensuráveis (numéricos) relativos à área de negócio que se pretenda analisar. Para esta organização possíveis métricas seriam: o volume de unidades vendidas ou os lucros obtidos.

Um **cubo virtual** é uma combinação de múltiplos cubos recorrendo à noção de estrutura lógica, à semelhança das vistas (*views*) dos sistemas de gestão relacional, em que se podem definir vistas sobre outras vistas e ou tabelas. Quando se cria um cubo virtual, seleccionam-se métricas e dimensões a partir do conjunto de métricas e dimensões dos cubos subjacentes. Para os utilizadores finais, um cubo virtual é visto como um único cubo. Um **cubo virtual** pode ainda ser parte de um único cubo expondo apenas um subconjunto de métricas e dimensões do cubo original.

Cada cubo é constituído por um conjunto de **células** que são definidas univocamente pela intersecção de todas as dimensões do cubo. Cada uma dessas células contém as métricas relativas à área de negócio em causa.

Um *data warehouse* utiliza um esquema de dados diferente dos sistemas transaccionais. Embora possam haver variantes, o esquema mais comum num *data warehouse* designa-se por **esquema em estrela** (*star-schema*) [Peterson00]. Este tipo de esquema de dados visa acima de tudo reduzir o número de ligações indirectas entre as diversas tabelas. Do ponto de vista físico, o conceito de cubo traduz-se no **esquema em estrela**.

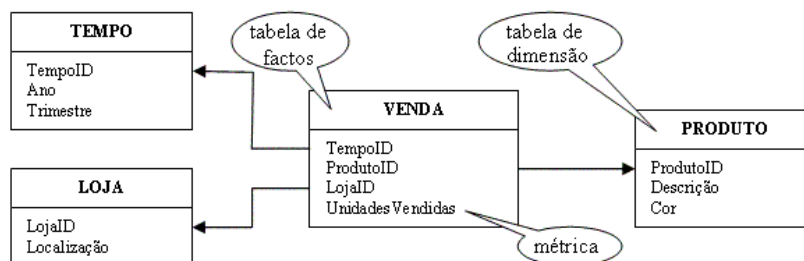


Figura 2.2 - Esquema em estrela exemplificativo

Tomemos como exemplo informação relativa às vendas de uma hipotética cadeia de lojas espalhadas por Portugal. A Figura 2.2 ilustra o respectivo esquema em estrela.

A tabela central do esquema designa-se de **tabela de factos** (neste caso a tabela “Venda”). As tabelas de factos contêm valores numéricos (métricas) que podem ser agregados e que neste exemplo corresponde ao número de unidades vendidas. As tabelas de dimensões correspondem às tabelas em redor da tabela de factos, tendo cada uma delas uma chave primária apontada por cada um dos campos da tabela de factos. Daí a designação de esquema em estrela. Neste exemplo temos como dimensões o Tempo, o Produto e a Loja. De referir que na maioria dos *data warehouses* existe a dimensão Tempo, pois, de uma forma geral, pretende-se analisar a evolução de qualquer aspecto do negócio ao longo do tempo.

Cada uma das dimensões está agrupada por diversos **níveis**. Por exemplo, na dimensão Tempo temos os níveis Ano e Trimestre. Estes dois níveis em conjunto definem uma **hierarquia**, que consiste em diversas relações pai-filho entre níveis da mesma dimensão.³

Finalmente designa-se de **membro** de uma dimensão, o valor guardado num nível dessa dimensão. Como exemplo teríamos para membros da dimensão Tempo no nível Ano os valores 2002 e 2003; e como membros da dimensão Tempo no nível Trimestre os valores 1, 2, 3 e 4.

Segundo este exemplo o conceito de cubo apresentado anteriormente traduzir-se-ia na imagem apresentada na Figura 2.3.

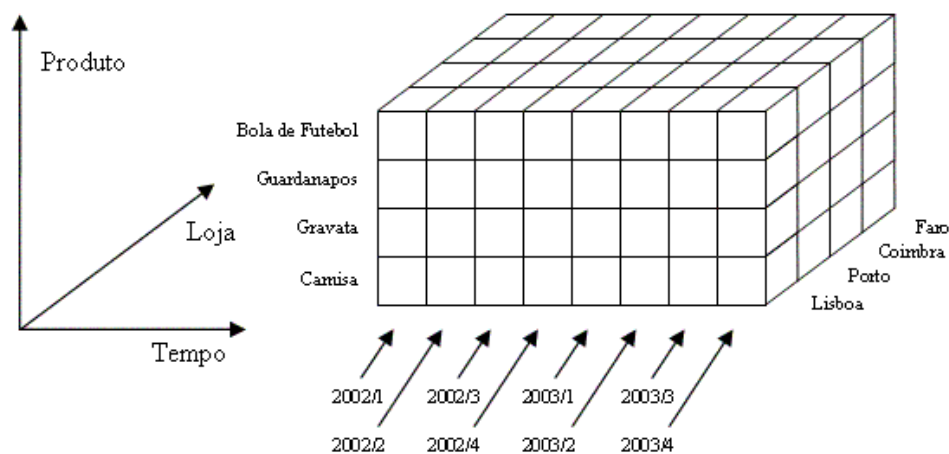


Figura 2.3 – Interpretação visual do cubo

³ Embora a mesma dimensão possa ter várias hierarquias esta funcionalidade não é suportada por alguns sistemas OLAP.

Uma vez que o cubo possui todos os dados relativos ao negócio de uma forma agregada, torna-se assim possível, e de uma forma rápida, responder a questões tais como: “Qual foi o volume de unidades vendidas de bolas de Futebol no ano de 2002?” ou “Qual foi o número de unidades vendidas de guardanapos na cidade de Lisboa nos últimos dois anos?”. Esta é a grande vantagem da abordagem OLAP: é possível realizar questões pertinentes do negócio e obter uma resposta de uma forma rápida [Dubler02].

2.1.4. Operações OLAP

Nesta secção irão ser descritas operações que podem ser executadas sobre dados multidimensionais. Embora muitas outras operações existam e possam ser realizadas sobre estruturas de dados multidimensionais, as seguintes operações são as mais conhecidas e recorrentes [Thomsen99]. Para que melhor se compreenda a forma como se realiza cada uma das operações, tomemos como exemplo o modelo de dados e o cubo apresentados na secção anterior (ver Figuras 2.2 e 2.3).

VENDAS			
TempoID	ProdutoID	LojaID	Unidades Vendidas
2	2	1	80
5	3	4	100
6	1	2	200
3	2	4	50
6	4	1	120
1	2	2	160
8	4	4	130
4	1	1	190
7	3	3	60
2	2	3	70
1	3	2	110
3	1	3	100
4	4	1	20
5	2	4	70

Tempo		
TempoID	Ano	Trimestre
1	2002	1
2	2002	2
3	2002	3
4	2002	4
5	2003	1
6	2003	2
7	2003	3
8	2003	4

Produto		
ProdutoID	Descrição	Cor
1	Bola de futebol	Branco
2	Guardanapos	Branco
3	Gravata	Preto
4	Camisa	Preto

Loja	
LojaID	Localização
1	Lisboa
2	Porto
3	Coimbra
4	Faro

Figura 2.4 – Tabela de factos e tabelas de dimensões com os respectivos valores

Este modelo de dados multidimensional é composto por uma tabela de factos e por três dimensões, sendo cada uma das dimensões representada por uma tabela. Neste exemplo guarda-se informação relativa às vendas de uma hipotética cadeia de lojas espalhadas por Portugal. Mais precisamente, guarda-se informação relativa ao número de unidades vendidas, tendo em conta a localização geográfica de cada loja, o período temporal e o respectivo produto em causa. De referir que a dimensão “Tempo” neste exemplo, é constituída por uma hierarquia com dois níveis: o nível “Ano” e o nível “Trimestre”.

Embora seja difícil a representação visual de cubos com mais de três dimensões eles existem. Neste exemplo, para facilitar a interpretação visual optou-se por construir o cubo com apenas três dimensões, como pode ser observado na Figura 2.3.

Seguidamente, é sobre esta informação disponível que irá recair a descrição de cada uma das operações de OLAP.

Operação *Roll Up*

As operações de *Roll Up* e de *Drill Down* são possivelmente as duas operações mais importantes no processamento analítico (OLAP). Ambas permitem aos utilizadores alterarem a granularidade com que os dados são consultados. A primeira, a operação de *Roll Up*, permite que o utilizador suba um nível de abstracção nos dados, ou seja que, agrupe os dados respeitantes a determinada dimensão um nível acima na sua hierarquia.

Todas as Lojas

	2002/1	2002/2	2002/3	2002/4	2003/1	2003/2	2003/3	2003/4
Bola de Futebol			100	190		200		
Guardanapos	160	150	50		70			
Gravata	110				100		60	
Camisa				20		120		130

Tabela 2.1 – Unidades vendidas em todas as lojas de Portugal



Todas as Lojas

	2002	2003
Bola de Futebol	290	200
Guardanapos	360	70
Gravata	110	160
Camisa	20	250

Tabela 2.2 – Resultado da operação de *Roll Up* sobre a dimensão Tempo aplicada à Tabela 2.1

Como pode ser observado pelo resultado na Tabela 2.2 foi aplicada a operação de *Roll Up* sobre a dimensão “Tempo” aos dados da Tabela 2.1. Assim, após a dita operação, os dados foram agregados relativamente à dimensão “Tempo”, ou seja, passou-se do nível Trimestre para o nível Ano. Como exemplo, temos que o número de bolas de futebol vendidas em todo o país durante o ano de 2002 resulta da soma do número de bolas de futebol vendidas em cada um dos trimestres desse mesmo ano ($290 = 100 + 190$).

Operação *Drill Down*

A operação de *Drill Down* traduz-se na operação inversa à operação de *Roll Up*, ou seja, permite descer um nível na análise de uma hierarquia de uma determinada dimensão. Desta

forma obtém-se uma visão mais granular da informação. Um possível exemplo da aplicação desta operação seria a passagem da Tabela 2.2 para os dados na Tabela 2.1.

Operação *Slice*

As operações de *Slice* e *Dice* correspondem à selecção de determinados membros num subconjunto de dimensões. Os seus nomes advêm precisamente do efeito de visualização causado no cubo por esta operação.

Loja=Faro

	2002/1	2002/2	2002/3	2002/4	2003/1	2003/2	2003/3	2003/4
Bola de Futebol			50		70			
Guardanapos					100			
Gravata								
Camisa								130

Tabela 2.3 – Resultado em tabela para a operação de *slice* aplicada à dimensão Loja=Faro

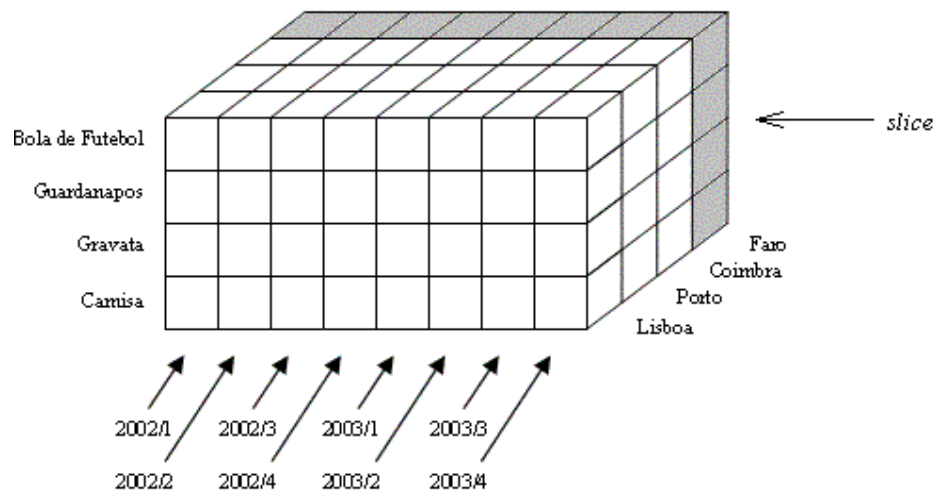


Figura 2.5 – Interpretação visual para a operação de *slice*

A Tabela 2.3 apresenta o resultado da aplicação da operação de *Slice* aos dados da Tabela 2.1. Nesta operação foi escolhido o membro “Loja=Faro” na dimensão Loja. Porém, outros membros de outras dimensões poderiam ter sido seleccionados. O efeito visual causado no cubo é a ideia de que é escolhida uma fatia do mesmo, como sugerido na Figura 2.5 (parte acinzentada), e daí o nome desta operação.

Operação *Dice*

A diferença entre a operação de *Dice* e a operação de *Slice* é que a primeira permite escolher mais do que um membro em cada dimensão, ao contrário da última que permite apenas a selecção de um único membro para cada dimensão.

		2002/1	2002/2	2002/3	2002/4	2003/1	2003/2	2003/3	2003/4
Bola de Futebol	Faro						200		
Guardanapos	Porto			50		70			
Gravata	Faro	160				100			
Camisa	Porto	110							130
	Faro								
	Porto								

Tabela 2.4 – Resultado em tabela da operação de *dice* para a dimensão “Loja=Faro ou Loja=Porto”

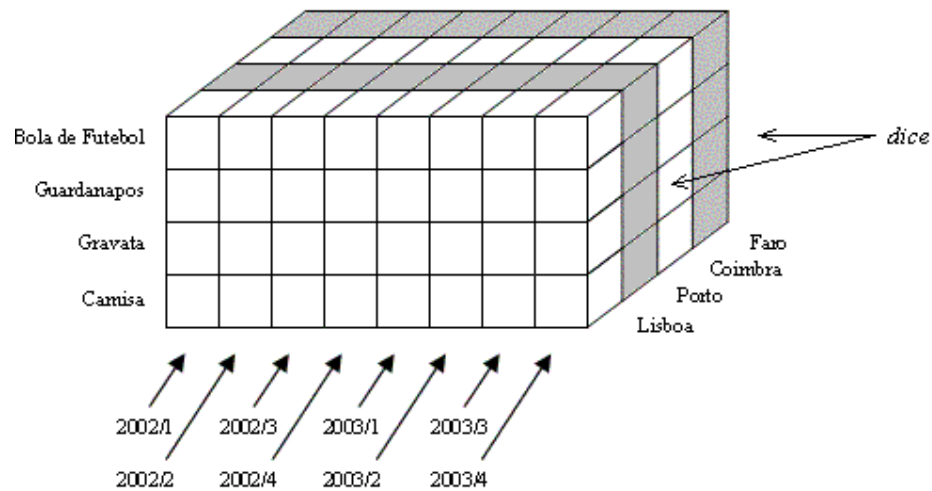


Figura 2.6 – Interpretação visual da operação de *dice*

Na Tabela 2.4 podem ser observados os resultados da aplicação da operação de *Dice* aos dados da Tabela 2.1. Na Figura 2.6 observa-se o efeito visual desta operação no cubo (selecção de duas fatias).

2.2. Armazenamento Multidimensional

Na modelação multidimensional existem três tipos distintos de armazenamento para os dados. Cada um dos tipos de armazenamento tem vantagens e desvantagens. Primeiramente abordamos cada um deles de um modo resumido e, de seguida, referimos algumas considerações acerca dos seus convenientes e inconvenientes. Por último, comparamos sucintamente os diferentes modos de armazenamento de dados multidimensionais.

2.2.1. MOLAP (*Multidimensional Data Storage*)

O modo MOLAP (*Multidimensional OLAP*) utiliza um tipo de armazenamento de dados criado especificamente para análises multidimensionais. Este modo trata os dados e as agregações da seguinte forma:

- Os dados são copiados da sua origem e armazenados numa estrutura multidimensional especializada para o cubo. Sempre que forem efectuadas *queries* ao cubo, os dados originais não são utilizados, uma vez que toda a informação está agora disponível nesta estrutura multidimensional;
- As agregações são armazenadas também nesta estrutura multidimensional do cubo.

2.2.2. ROLAP (*Relational Data Storage*)

O modo ROLAP (*Relational OLAP*) utiliza as estruturas de uma base de dados relacional para guardar as agregações do cubo. Assim, os dados e as agregações são tratados do seguinte modo:

- Os dados são mantidos na sua origem. Sempre que forem realizadas *queries*, os dados são obtidos a partir da sua origem, que em geral é um SGBD relacional.
- As agregações são guardadas na base de dados relacional sob um conjunto adicional de tabelas. A informação nestas tabelas é obtida para dar resposta às *queries* MDX realizadas ao cubo.

2.2.3. HOLAP (*Hybrid Data Storage*)

Tal como o próprio nome indica, o modo HOLAP combina características tanto do modo MOLAP como do modo ROLAP. Este modo trata os dados da mesma forma que o modo ROLAP e as agregações da mesma forma que o modo MOLAP:

- Os dados são deixados na sua origem e são aí acedidos sempre que forem efectuadas *queries* ao cubo em questão.
- As agregações são mantidas na estrutura multidimensional especializada para o cubo.

2.2.4. Comparação entre MOLAP, ROLAP e HOLAP

A escolha do tipo de armazenamento dos dados tem um impacto relevante em três aspectos: tempo de processamento do cubo, espaço alocado para o cubo e velocidade de consulta do cubo. É basicamente sobre estes três aspectos que deve recair a nossa atenção na escolha do tipo de armazenamento a utilizar.

Os principais factores relevantes na escolha do modo de armazenamento MOLAP são os seguintes:

- As operações de consulta ao cubo são bastante mais rápidas, mesmo que não sejam criadas agregações;
- MOLAP utiliza geralmente mais espaço em disco uma vez que os dados são copiados da sua origem;
- O cubo, segundo o modo MOLAP, pode ser consultado mesmo quando a origem dos dados não está disponível, uma vez que toda a informação está armazenada no próprio cubo;
- Este tipo de armazenamento é provavelmente a melhor escolha, a menos que exista um elevado conjunto de dados que não seja passível de ser consultado no futuro com frequência. Neste caso, não faz sentido a utilização do tempo de processamento e do espaço de armazenamento necessários na cópia dos dados para o formato multidimensional.

Os principais factores que afectam a escolha do modo ROLAP são os seguintes:

- As operações de consulta aos cubos são mais lentas comparativamente com o modo MOLAP, e praticamente o mesmo que no modo HOLAP.
- O tempo de processamento é bastante maior, especialmente em níveis de agregação mais elevados.
- O cubo não pode ser consultado, a menos que haja ligação à fonte de dados.
- Caso não se utilizem muitas agregações, o modo ROLAP praticamente não utiliza nenhum espaço de armazenamento.
- As agregações neste tipo de armazenamento ocupam muito mais espaço que as agregações nos outros dois modos (MOLAP/HOLAP).

Os factores que afectam a escolha do modo HOLAP são os seguintes:

- Quando se pretende aceder aos níveis mais baixos dos dados este modo de armazenamento comporta-se exactamente como o modo ROLAP. No acesso às agregações, este modo de armazenamento comporta-se como o modo MOLAP;

- Só é possível consultar o cubo se houver ligação à origem dos dados;
- HOLAP é a melhor escolha quando não se pretende aceder aos níveis de dados mais baixos. Além disso, neste último caso, é possível poupar espaço de armazenamento e tempo de processamento.

2.3. Análise de Ferramentas de *Data Warehousing*

2.3.1. Microsoft SQL Server 2000 Analysis Services

O Microsoft SQL Server 2000 Analysis Services, doravante apenas designado por Analysis Services, é o servidor que acompanha o Microsoft SQL Server 2000 e que se destina às áreas de *data warehousing* e de *data mining*. Nesta secção descreve-se as principais características deste servidor que contribuíram para a sua adopção nesta dissertação, dando ênfase às funcionalidades vocacionadas para a área de *data warehousing*.

Os Analysis Services oferecem três modos distintos para o armazenamento multidimensional da informação: o modo MOLAP que guarda os dados e as agregações em estruturas multidimensionais; o modo ROLAP que armazena os dados juntamente com as agregações numa base de dados relacional; e o modo HOLAP, que armazena os dados do cubo numa base de dados relacional e as agregações numa estrutura multidimensional.

Um dos aspectos fundamentais dos Analysis Services é a sua facilidade de utilização graças ao conjunto de *wizards* e editores que acompanham a ferramenta. De entre todos os *wizards* disponíveis, que endereçam as diferentes etapas de construção de uma *data warehouse*, destacam-se os que permitem a construção de cubos, a construção de dimensões e a construção de cubos virtuais. Para além deste conjunto de *wizards* existe ainda um conjunto de ferramentas que permite a edição dos objectos criados: o *Cube Editor* (ver Figura 2.7), o *Dimension Editor* e o *Virtual Cube Editor*. De referir ainda o facto dos Analysis Services disponibilizarem uma interface bastante rica para consulta de metadados relativos a cubos, dimensões, hierarquias, níveis, etc.

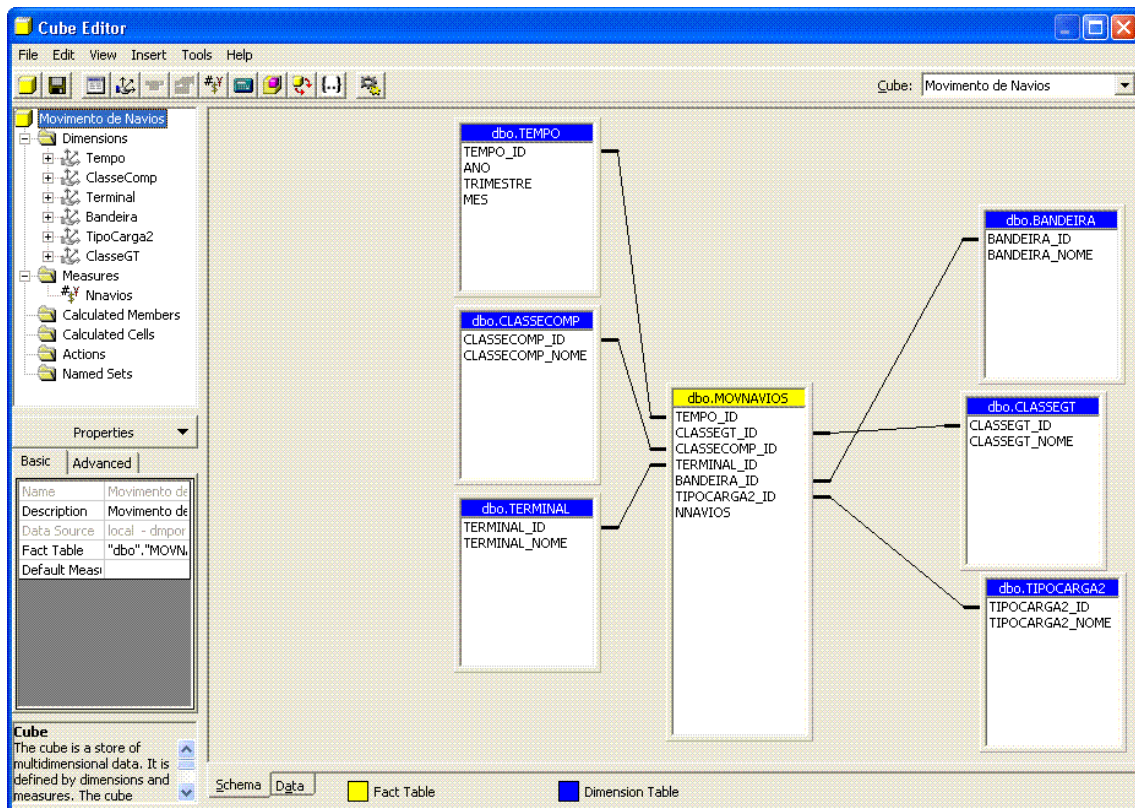


Figura 2.7 – Exemplo da edição de um cubo através da ferramenta *Cube Editor*

Esta plataforma de *data warehousing* e de *data mining* foi desenhada de forma a facilitar a sua integração com outras aplicações. Algumas das principais potencialidades desta plataforma são:

- A compatibilidade com os *ActiveX Data Objects* (ADO) e com a sua extensão para objectos multidimensionais (ADO MD).
- O facto de, através do modelo de objectos DSO (*Decision Support Objects*), ser possível fazer quase por completo a gestão (de forma programática) do *data warehouse*, como por exemplo, criar e modificar cubos, dimensões, níveis.
- Ser possível criar extensões que podem ser integradas com o *Analysis Manager* (gestor de interface gráfica para os *Analysis Services*), tais como caixas de diálogo ou novos *wizards*.

Um problema comum às bases de dados OLAP é a explosão de dados (*data explosion*), originado nomeadamente pelos seguintes factores: espaço necessário aos dados e índices; espaço dedicado a células vazias; e espaço dedicado às agregações. A arquitectura dos *Analysis Services* contorna este problema através de uma compressão eficiente dos dados, não armazenamento de células vazias, e de uma pré-computação das agregações. A pré-computação das agregações é facilitada por um *wizard* (*Storage Design Wizard*) que permite escolher o nível de agregação pretendido. Mediante o nível escolhido assim resultará um

melhor ou pior desempenho no acesso aos dados. Um maior nível de agregação implicará um maior tempo de processamento das agregações, um maior espaço ocupado em disco e um menor tempo no acesso aos dados, e vice-versa. A Figura 2.8 apresenta um exemplo da utilização do *Storage Design Wizard* para um nível de agregação de 50%.

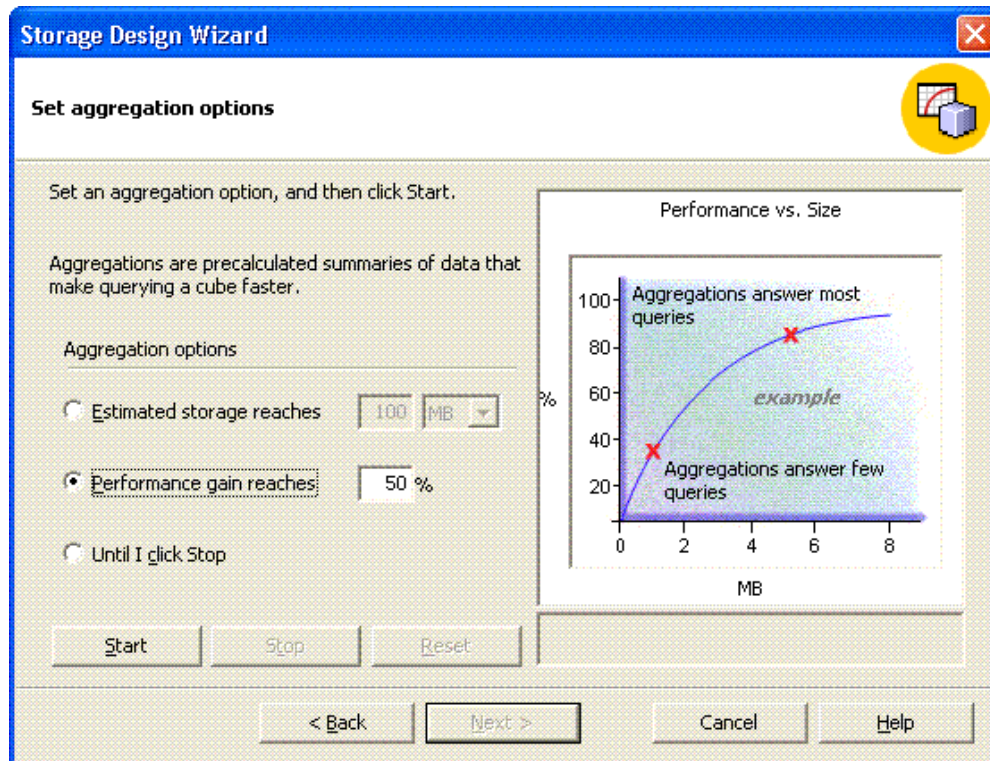


Figura 2.8 – Exemplo da pré-computação das agregações através do *Storage Design Wizard*

Outras funcionalidades consideradas importantes nos Analysis Services são apresentadas em seguida:

- Possibilidade de aceder a dados localizados em muitos tipos de fontes de dados (incluindo fontes ODBC e OLE DB). Para além disso é possível definir diferentes fontes de dados para o mesmo *data warehouse*.
- Possibilidade de criar partições que podem conter apenas uma porção dos dados relativos a um cubo. As partições permitem distribuir os dados de um cubo por diferentes servidores. Para além de um aumento de escalabilidade, esta opção proporciona melhorias no desempenho do processamento dos cubos e na realização das análises.
- Possibilidade de visualização directa (*browsing*) dos dados contidos nos diferentes cubos.
- Disponibilização de um *wizard* para actualização incremental de informação.
- Disponibilização de uma ferramenta para consulta estatística relativamente aos cubos consultados.

- Existência de dois níveis de autenticação: a nível do Windows 2000/NT e a nível do IIS (*Internet Information Services*).
- Possibilidade de atribuição de papéis (*roles*) específicos a diferentes utilizadores para acesso a diferentes cubos. Podem ainda ser definidos certos níveis de segurança a células e membros específicos.
- Possibilidade de definição de múltiplas hierarquias para a mesma dimensão.
- Possibilidade de definição de métricas derivadas (*calculated measures*) e membros derivados (*calculated dimension members*) através de expressões multidimensionais (MDX), fórmulas matemáticas e funções definidas pelo utilizador.

Como se pode verificar esta plataforma de *data warehousing* possui muitas funcionalidades, satisfazendo plenamente os requisitos desta dissertação.

2.4. Análise de Ferramentas de *Reporting*

Um dos primeiros passos na abordagem ao problema identificado no primeiro capítulo desta dissertação, consistiu no estudo de algumas ferramentas comerciais para criação de relatórios. Das várias ferramentas disponíveis no mercado optou-se por estudar duas delas: o Crystal Reports e os Reporting Services. Ambas as ferramentas são actualmente líderes no mercado, com grande reputação, fiáveis e bastante versáteis. O Crystal Reports, da empresa Business Objects, tem grande tradição, nomeadamente devido à sua inclusão nos pacotes Visual Studio. Os Reporting Services, embora lançados recentemente pela Microsoft, são já uma referência na área de *reporting*.

Embora ambas as ferramentas constituíssem à partida duas opções válidas, a escolha recaiu sobre os Reporting Services devido às seguintes dificuldades encontradas durante a análise dos Crystal Reports relativamente aos métodos disponibilizados pela ferramenta para integração de relatórios com aplicações:

- **RDC (*Report Designer Component*)**: não obstante a versão Crystal Reports Visual Studio .Net possuir dois *viewers* (o *Windows Form Viewer* e o *Web Form Viewer*), nenhum dos dois permite utilizar relatórios construídos pelo método RDC.
- **CrystalDecisions.CrystalReports.Engine**: este motor de relatórios que integra as versões *Advanced* e *Developer* do Crystal Reports Visual Studio .Net não contempla as funcionalidades de criação de relatórios de forma dinâmica.

- **RAS (*Report Application Server*)**: embora este método permita a criação dinâmica de relatórios não permite o acesso a fontes de dados OLAP, nomeadamente aos Analysis Services. O método apenas permite a visualização de relatórios estáticos (isto é, definidos à partida) baseados em dados multidimensionais.

Esta secção concentra-se então na descrição da arquitectura e principais funcionalidades da ferramenta de *reporting* utilizada no desenvolvimento desta dissertação, os Reporting Services, fazendo referência às principais vantagens que conduziram à sua adopção.

2.4.1. Microsoft SQL Server 2000 Reporting Services

Os Reporting Services constituem a plataforma de *reporting* da Microsoft⁴. Através desta plataforma é possível criar relatórios em tabelas, matrizes ou gráficos acedendo a bases de dados relacionais ou multidimensionais. A plataforma tem a particularidade de permitir a visualização de relatórios através de uma ligação Web. Os Reporting Services são compostos por um conjunto de serviços, ferramentas e API's (*Application Programming Interface*) que permitem a construção, gestão e publicação de relatórios. Nas subsecções seguintes irá ser descrita a arquitectura dos Reporting Services, bem como as principais funcionalidades que esta ferramenta de *reporting* suporta.

Arquitectura

Os Reporting Services possuem uma arquitectura modular, como pode ser observado em seguida na Figura 2.9. A plataforma tem por base o Report Server cuja função consiste em suportar toda e qualquer interacção dos utilizadores. Para além do Report Server, existem ainda um conjunto de módulos, que conferem à plataforma um carácter extensível.

Para além da plataforma suportar um conjunto diversificado de fontes de dados, tais como, SQL Server, Oracle, OLE DB ou ODBC, esta arquitectura dos Reporting Services está desenhada de forma a suportar novos tipos de fontes de dados.

O mesmo se passa relativamente aos formatos de *output*. Para além dos formatos de *output* disponibilizados pela plataforma (HTML, PDF, Microsoft Excel), os programadores podem criar as suas próprias extensões de *rendering* de modo a tirar vantagem das capacidades de outros tipos de dispositivos.

⁴ Os Reporting Services surgiram em Janeiro de 2004 durante o decurso desta investigação

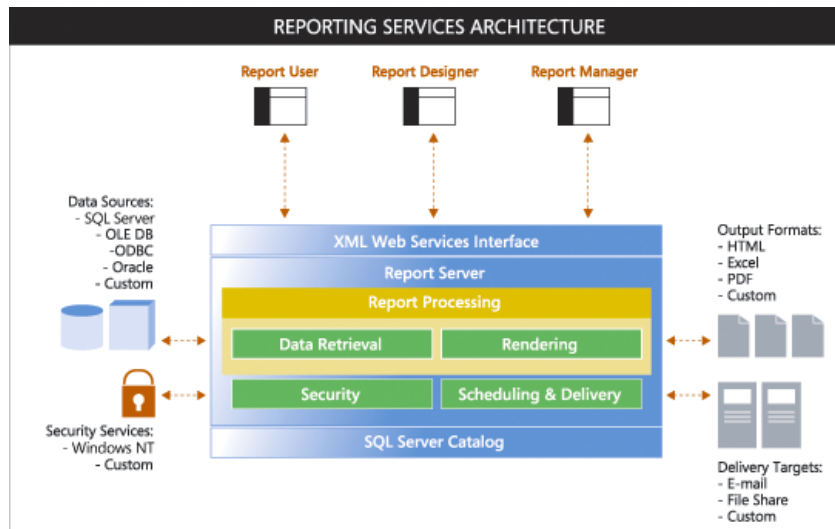


Figura 2.9 – Arquitetura dos Reporting Services

Quanto à autenticação suportada pela arquitectura baseia-se na autenticação Windows, embora se possam definir outros tipos de segurança.

No que toca à distribuição de relatórios, esta arquitectura encontra-se preparada para efectuar uma distribuição via *e-mail*, embora outros tipos de distribuição possam ser programados.

O **Report Server**, como componente principal da arquitectura, funciona como um serviço normal num ambiente Windows.

A **base de dados** do Report Server encontra-se no SQL Server e armazena toda a informação referente aos Reporting Services. Esta informação passa pelo registo das definições dos relatórios, dos metadados associados aos relatórios, relatórios em *cache*, *snapshots*, configurações de segurança, dados encriptados, configurações de *scheduling* e outras informações referentes a extensões aos Reporting Services.

Esta base de dados apenas é acessível através do *Report Server*, ou seja, mesmo utilizando o *Report Manager*, o *Report Designer* ou outros “utilitários *command-line*”, estamos a utilizar interfaces programáticas de acesso à base de dados que comunicam com o *Report Server* de uma forma transparente.

O **Report Manager** é uma aplicação Web que acompanha a plataforma com o intuito de gerir e aceder relatórios através de uma interface gráfica. A partir do Report Manager é possível realizar as seguintes operações: (1) Visualizar, procurar e subscrever relatórios; (2) Criar e gerir pastas, *linked reports*, ligações a fontes de dados e subscrições de relatórios; (3) Criar e

modificar propriedades e parâmetros dos relatórios; e (4) Gerir definições de papéis que controlam o acesso a relatórios e pastas por diferentes utilizadores.

O **Report Designer** é uma ferramenta que permite a criação e a publicação de relatórios de uma forma simples e cómoda. Esta ferramenta acompanha o Visual Studio .Net 2003 e utiliza uma interface gráfica bastante prática.

Há ainda a realçar a existência de um componente desenvolvido pela Microsoft, designado de **Report Viewer**, para visualização de relatórios em páginas Web. Através deste componente podemos de uma forma fácil e prática realizar a exportação de relatórios para diversos formatos, efectuar operações de *drill down*, entre outras funcionalidades.

Funcionalidades

Um relatório segundo os Reporting Services pode conter os seguintes tipos de objectos: linhas, rectângulos, caixas de texto, imagens, sub-relatórios, listas, matrizes, tabelas e gráficos. A plataforma suporta ainda um outro tipo de objecto, designado por *custom report item*, indicado para os casos em que se pretende inserir no relatório um outro tipo de objecto que não os disponibilizados pela plataforma. Neste caso, é necessário haver uma aplicação ou servidor específico para o efeito que interprete e faça o *rendering* do objecto por nós definido.

É possível através desta plataforma de *reporting* definir campos que indiquem sub-totais ou campos que definem totais. Embora a função de soma seja a função mais comum, outras funções podem ser definidas relativamente a um conjunto de valores, tais como, a média, mediana, desvio-padrão, entre outras funções matemáticas. Esta funcionalidade é bastante útil nos casos em que os relatórios contêm diversas páginas.

Esta ferramenta de *reporting* possui a capacidade de efectuar operações OLAP em dados agregados, ou seja, entra dentro do domínio do *reporting* interactivo. A capacidade de realizar a operação de *drilling* segundo diversos níveis de granularidade da informação constitui uma mais valia para a ferramenta. Por exemplo, podemos navegar na hierarquia ano-trimestre-mês caso estes dados constem no *dataset*⁵ definido.

Os Reporting Services contam com um conjunto diversificado de formatos de exportação de relatórios. É possível exportar um relatório para o Microsoft Excel (formato XLS), para o Acrobat Reader (formato PDF), para um ficheiro de XML ou para o componente PivotTable

⁵ Um *dataset* consiste no conjunto de dados que suporta a informação a constar no relatório

que pertence à gama de componentes *Web* disponíveis no Microsoft Office (*Office Web Components* – OWC).

Os Reporting Services possuem várias técnicas ou funcionalidades de interacção com o utilizador. Uma das técnicas já foi anteriormente referida - as operações OLAP. As outras técnicas que permitem a interacção com o utilizador são a capacidade de parametrização do relatório, os filtros, os *links* e o mapa do documento (*document map*).

Como se pode verificar esta plataforma de *reporting* possui muitas funcionalidades no que diz respeito à construção e visualização de relatórios, satisfazendo plenamente os requisitos exigidos por esta investigação relativamente ao desenho e à construção de relatórios. Para além das vantagens inerentes a tantas funcionalidades, os Reporting Services têm a particularidade da definição dos seus relatórios ser realizada segundo um formato XML designado por RDL (*Report Definition Language*). O facto dos relatórios serem definidos segundo uma linguagem aberta, o XML, permite aos seus utilizadores definirem novos elementos de acordo com as necessidades do seu negócio. Este aspecto será abordado em maior profundidade no Capítulo 5.

“As pessoas que sobem neste mundo são as pessoas que se levantam e procuram o que querem e que, se não o encontrarem, fazem-no”

George Bernard Shaw

Capítulo 3.

Caso de Estudo: Gestão Portuária – DMPORT

Sumário

Neste capítulo explica-se de forma sucinta o contexto em que esta dissertação foi desenvolvida. Inicialmente, explica-se a área de negócio deste trabalho e contextualiza-se o caso de estudo. De seguida descreve-se o modelo relacional de dados (modelo original) sobre o qual recaiu a investigação. Descrevem-se ainda as principais funcionalidades existentes no sistema legado estudado. Posteriormente explicam-se as motivações e o problema que conduziram ao desenvolvimento do trabalho bem como os seus principais requisitos.

3.1. Introdução ao Caso de Estudo

Esta dissertação foi desenvolvida no contexto de sistemas de apoio à decisão, tendo sido validada com base num caso de estudo no âmbito dos transportes marítimos, designadamente no âmbito da gestão de tráfegos portuários e correspondente informação envolvida. De forma a facilitar a integração do leitor no caso de estudo descrever-se-á em seguida, de modo sucinto, o funcionamento típico de um sistema de “Gestão Portuária”.

A actividade portuária identifica-se por um conjunto de interacções entre diferentes tipos de interlocutores ou entidades. O universo em análise é definido por um conjunto de acções e entidades a si associadas.

Nos portos podem ocorrer diferentes tipos de tráfegos, como por exemplo, tráfego de navios comerciais ou de pesca, tráfego de mercadorias, tráfego de passageiros, e tráfego de contentores. Cada tipo de tráfego requer um tratamento particular. No entanto, esta dissertação apenas assenta sobre os movimentos de mercadorias e de passageiros, para além de alguns indicadores específicos de gestão, tais como, análise de ocupação de cais.

Como pode ser observado, na Figura 3.1, um porto é constituído fisicamente por um conjunto de **terminais**, os quais agregam diferentes **cais**.

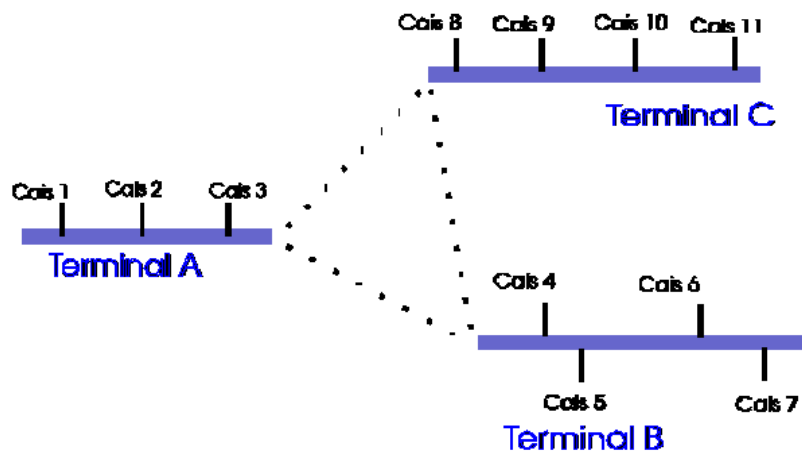


Figura 3.1 – Organização de um porto em terminais e cais (adaptado de [Planibase94])

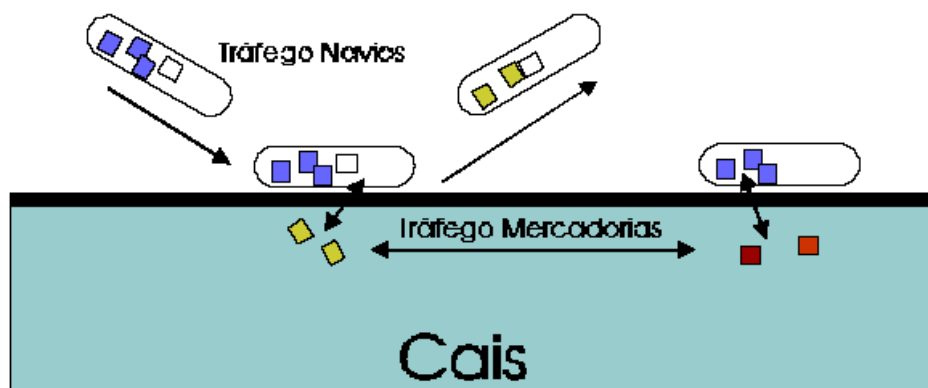


Figura 3.2 – Tráfego de navios e de mercadorias (adaptado de [Planibase94])

Os dois fluxos mais importantes de um porto comercial são precisamente o **tráfego de navios** e o **tráfego de mercadorias**, tal como ilustrado na Figura 3.2. Um dos objectivos principais de um porto comercial consiste em maximizar o tráfego de navios e consequente tráfego de

mercadorias, de forma a rentabilizar os seus recursos, como por exemplo, área de cais e/ou área de carga/descarga.

Sempre que um navio entra num porto é registado esse acontecimento num documento designado de “Processo de Entrada do Navio”. A entrada de um navio num porto origina geralmente operações de carga/descarga, embora o navio possa também acostar por outros motivos diversos, como por exemplo, para abastecimento de mantimentos/combustível, ou por doença.

Para o cálculo de certos indicadores, tais como a Taxa de Ocupação de Cais, é pertinente o registo da hora e data de **acostagem/desacostagem** do navio, assim como do registo do local (i.e. terminal/cais) respectivo.

De todos os tráfegos de mercadorias que possam ocorrer nos portos, existe um tipo particular, o movimento de **contentores**. Associado a cada contentor existe um conjunto de informação que é necessária registar: o tamanho do contentor, se está cheio ou vazio, as mercadorias que transporta, etc. Um tratamento semelhante ao dos contentores é o das **unidades móveis (Ro-Ro's)**.

Tendo em conta todo este conjunto de acções desenroladas num porto, convém descrever as entidades que interferem nestes fluxos. A Figura 3.3 sugere algumas dessas entidades que, directa ou indirectamente, estão envolvidas nos processos de gestão portuária.

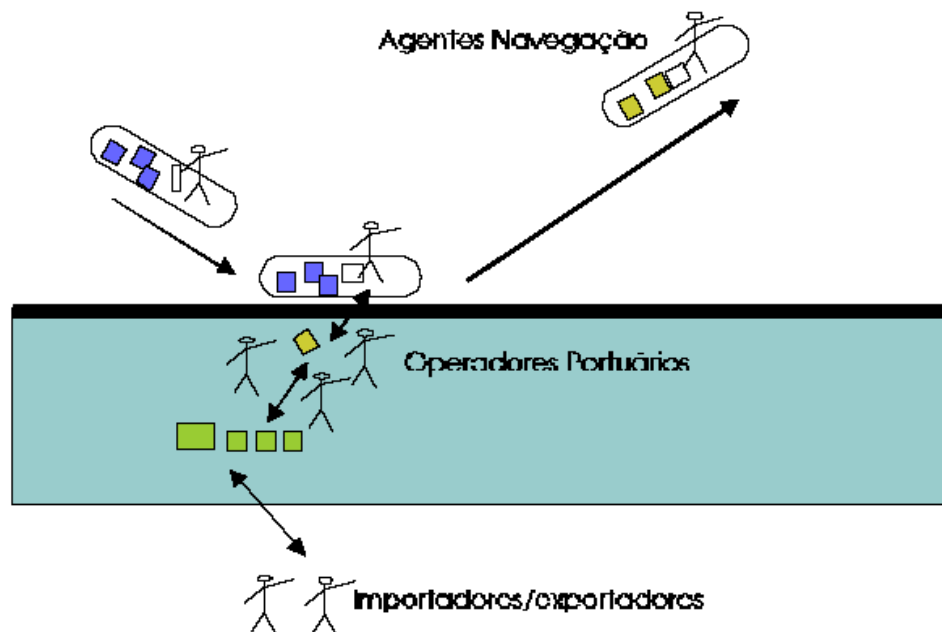


Figura 3.3 – Interação entre as entidades envolvidas na gestão portuária (adaptado de [Planibase94])

Existem três tipos essenciais de entidades envolvidas no processo de gestão portuária:

- **Importadores/Exportadores:** responsáveis directos pelo envio ou recepção das diversas mercadorias (na falta destes são substituídos pelos **agentes transitários**).
- **Agentes de Navegação:** responsáveis pelo fretamento dos navios, que estabelecem uma forte relação com os donos (ou armadores) dos navios.
- **Operadores Portuários:** responsáveis pela carga ou descarga de mercadorias.

Existem algumas notas que convém aqui referir:

- Um dado navio pode transportar mercadorias de diferentes importadores/exportadores, as quais podem ser descarregadas por diferentes operadores portuários;
- Um movimento de navio pode ter no máximo dois agentes de navegação associados, um associado ao processo de entrada (chegada do navio ao porto), e outro associado ao processo de saída (partida do navio) – embora a situação mais habitual seja a existência de um único agente de navegação associado a ambos os processos [Planibase94].

3.2. Modelo de Dados e Funcionalidades Anteriores

Tal como foi referido na secção 3.1 esta dissertação foi desenvolvida no contexto de transportes marítimos, designadamente administrações portuárias. O ponto de partida desta dissertação teve por base um produto designado de SITEP (Sistema de Tratamento Estatístico Portuário). O SITEP é um produto informático desenvolvido pela empresa PlaniBase para a área dos transportes marítimos (designadamente para as administrações portuárias).

O SITEP integra-se plenamente neste conjunto de sistemas considerando-se um produto para tratamento estatístico portuário do movimento comercial de navios, de mercadorias e de passageiros. Um aspecto relevante do sistema SITEP é a concordância da informação gerida com a directiva de normalização de estatísticas de transportes marítimos de mercadorias e de passageiros exigida pelo EUROSTAT – União Europeia [EUROST94], designada neste capítulo simplesmente de Directiva.

Um dos aspectos importantes deste sistema consiste na produção de um conjunto de listagens a ser enviadas ao EUROSTAT. Segundo a Directiva do EUROSTAT todos os portos comerciais de mercadorias e de passageiros dos países da União Europeia são obrigados a

fornecer um conjunto de informações estatísticas com periodicidade trimestral e anual.⁶ Alguns dos quadros de informação exigidos pela Directiva da União Europeia compreendem a seguinte informação:

- Total de mercadorias movimentadas (ton.) segundo o tipo de movimento (entrada/saída), porto de origem/destino, tipo de carga e costa marítima (A1).
- Total de mercadorias movimentadas (ton.) segundo o tipo de movimento, porto de origem/destino, tipo de carga detalhada e costa marítima (A2).
- Total de mercadorias e passageiros movimentados segundo o tipo de movimento (A3).
- Total de mercadorias movimentadas (ton.) segundo o tipo de movimento, porto de origem/destino, tipo de carga, agregação de mercadoria e costa marítima (B1).
- Movimento de navios (número de navios e *deadweight*) segundo o tipo de navio, classe de *deadweight* e tipo de movimento (F1).

Cada um dos quadros é identificado univocamente por dois caracteres (uma letra e um algarismo). Como exemplo de um desses quadros apresenta-se na Figura 3.4 o quadro referente ao movimento de navios segundo o tipo de movimento, tipo de navio e respectiva classe de *deadweight* (F1), produzido pelo sistema SITEP.

The screenshot shows a window titled 'C:\Admport\SITEP.EXE' with a menu bar 'Estatística Portuaria/Comercial <SITEP/C 3.3> - Demo - PB 03/03/2004'. The main content is a table titled '**** Consulta de F1 - Mov. Navios, por Tipo e Deadweight ****'. The table has six columns: 'Código Tabela', 'Porto Declarante', 'Direcção', 'Tipo de Navio (2)', 'Classe de Deadweight', and 'Nº de Navios'. The data is as follows:

Código Tabela	Porto Declarante	Direcção	Tipo de Navio (2)	Classe de Deadweight	Nº de Navios
F1	PTDDD	1	10	04	3
F1	PTDDD	2	10	04	3
F1	PTDDD	1	10	05	1
F1	PTDDD	2	10	05	1
F1	PTDDD	1	10	06	2
F1	PTDDD	2	10	06	2
F1	PTDDD	1	10	07	1
F1	PTDDD	2	10	07	1
F1	PTDDD	1	33	00	24
F1	PTDDD	2	33	00	24
F1	PTDDD	1	33	01	2
F1	PTDDD	2	33	01	2
F1	PTDDD	1	33	02	1
F1	PTDDD	2	33	02	1

At the bottom of the window, there is a status bar with the text: 'Remove L0ck Procura R.Todos ESC-Sai'.

Figura 3.4 – Quadro F1 – Mov. Navios por Tipo de Operação, Tipo de Navio e Classe de Deadweight

A informação processada pelo SITEP na produção dos quadros para o EUROSTAT baseia-se em quatro tipos de factos que são registados directamente pelos utilizadores do sistema ou transferidos a partir da informação de sistemas operacionais designados por “centros de

⁶ Em Portugal o organismo responsável pela agregação e tratamento da informação estatística portuária é o Instituto Nacional de Estatística (INE)

despacho de navios”: o movimento de navios; as acostagens; o movimento de mercadorias e o movimento de passageiros. Estes factos são estruturados através de um conjunto de relações que podem ser avaliadas pela observação do diagrama UML apresentado na Figura 3.5.

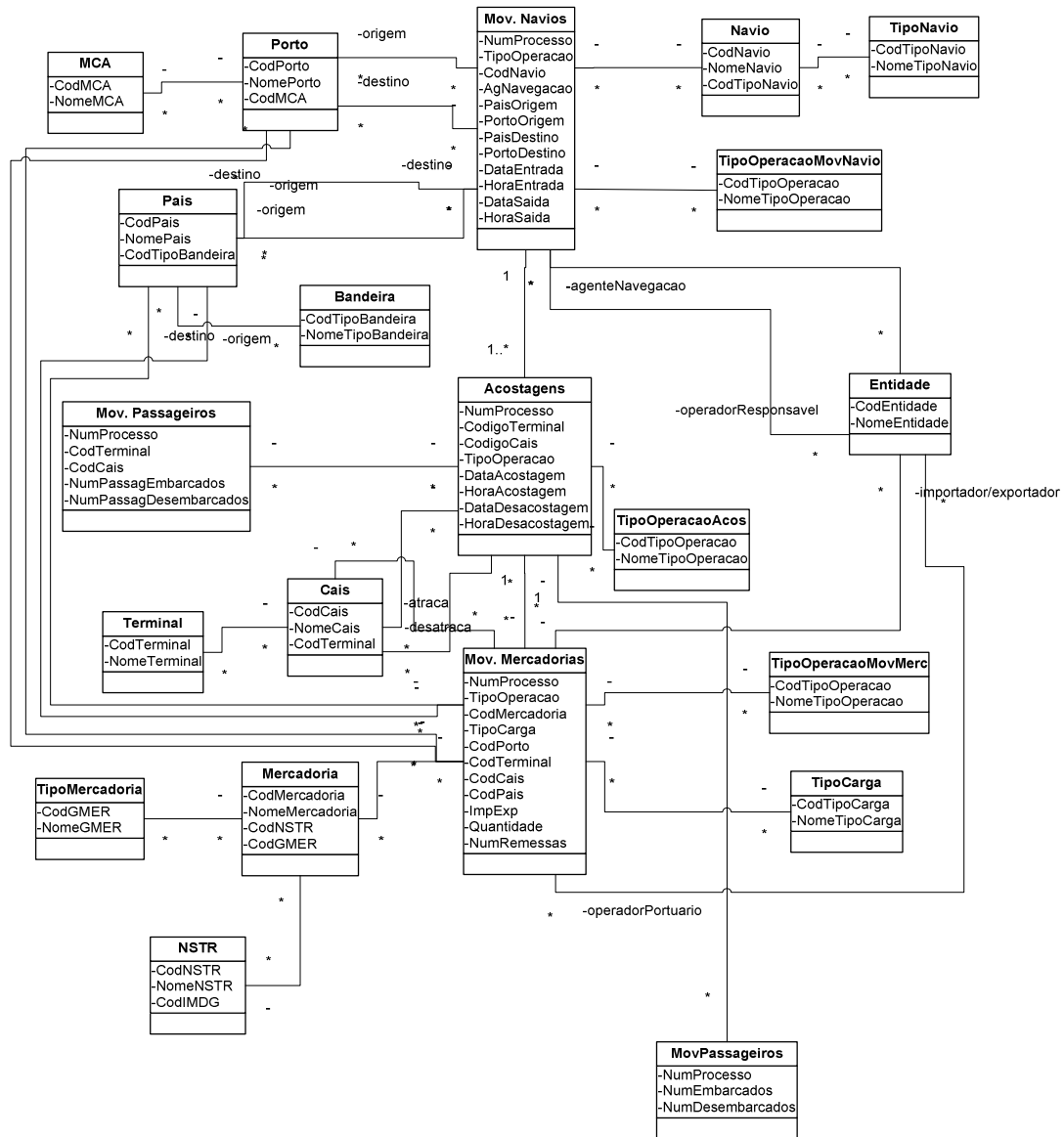


Figura 3.5 – Informação operacional de suporte às operações portuárias

Para além da produção dos quadros exigidos pelo EUROSTAT, o sistema SITEP permite também a exportação dos dados para ficheiros de formato DBF. DBF é um formato *standard* de ficheiros de base de dados em ambientes *desktop*.

3.3. Requisitos com Nova Configuração

O objectivo fulcral deste trabalho consiste em providenciar análises livres OLAP e de *reporting* e um conjunto de relatórios pré-configurados relativamente a informação de negócio gerida no contexto de uma administração portuária.

A metodologia utilizada para que o objectivo deste trabalho fosse atingido compreendeu uma série de tarefas que são sucintamente descritas de seguida.

A **primeira tarefa** passou por analisar de uma forma cuidada a informação de negócio gerida no contexto de uma administração portuária. Esta análise compreendeu um estudo aprofundado do modelo de dados subjacente à informação gerida. Este estudo teve como apoio o manual do sistema SITEP (Sistema de Tratamento Estatístico Portuário) [Planibase94].

Numa **segunda tarefa** foram definidos os relatórios considerados mandatórios. Com estes mesmos relatórios identificados tornou-se possível definir os modelos multidimensionais necessários que constituiriam o *data mart*.

Numa **terceira tarefa**, e certamente a mais árdua em termos de tempo e esforço de trabalho, foram definidas uma série de regras de transformação de dados, que são descritas no Capítulo 4, que têm por objectivo realizar a passagem da informação descrita segundo um modelo relacional para o modelo multidimensional.

Numa **quarta** e última tarefa foi construído um protótipo que possibilitasse as análises livres OLAP e de *reporting*, bem como a definição do conjunto de relatórios mais utilizados.

Em consequência da definição dos relatórios mandatórios chegou-se à conclusão que eram necessários 16 cubos para armazenar a informação necessária. Os cubos necessários a serem construídos são os seguintes:

- Cubos com informação exigida pelo Eurostat (12 cubos)
- Cubos com registos de movimentos (2 cubos)
- Cubos com informação sobre indicadores (2 cubos)

A título exemplificativo apresenta-se, em seguida, um dos cubos com informação exigida pelo Eurostat, a saber, o cubo EuroF1 (Movimento de Navios por Tipo e Classe de *Deadweight*).

EuroF1
Tempo
Tipo de Operação de Navio
Tipo de Navio
Classe de Deadweight
Nº Navios
Total Deadweight

Figura 3.6 – Cubo EuroF1 (Movimento de Navios por Tipo e Classe de *Deadweight*)

Como pode ser observado na Figura 3.6 o cubo EuroF1 contém quatro dimensões (Tempo, Tipo de Operação de Navio, Tipo de Navio e Classe de *Deadweight*) e duas métricas (Nº Navios e Total *Deadweight*).

No Apêndice A.1 podem ser consultados todos os cubos que foram construídos.

“Aproveitar um bom conselho requer mais sabedoria do que dá-lo”

John C. Collins

Capítulo 4.

Armazenamento e Transformação de Dados

Sumário

Neste capítulo irá ser abordada uma das tarefas mais importantes de um projecto de *data warehousing* – a tarefa do processo de transformação de dados -, usualmente chamada de ETL (*Extraction, Transformation, Loading*) [SAS-].

Começa-se por explicar em que consiste esta tarefa e quais os aspectos mais relevantes a ter em consideração para que se possa obter sucesso no processo de transformação de dados. De seguida faz-se uma breve abordagem à tecnologia utilizada pela Microsoft, o DTS (*Data Transformation Services*) [Larsen00], aplicada ao caso de estudo, e introduz-se a noção de “*package*” neste âmbito. Posteriormente, explica-se como foram desenvolvidos os *packages* necessários à implementação desta tarefa. Por último são apresentadas as dificuldades enfrentadas durante a análise do caso de estudo e quais as soluções adoptadas.

4.1. Introdução

O processo de transformação de dados, geralmente designado de ETL (*Extraction, Transformation, Loading*) é aquele que tipicamente requer a maior parte do tempo de desenvolvimento (podendo chegar a 50%) no ciclo de implementação de um projecto de *data warehousing* [DataWarehousing-]. Uma razão para este facto reside na necessidade de se

localizar a origem dos dados, entender as regras de negócio subjacentes escolhendo as colunas adequadas, bem como entender os modelos físico e lógico da informação [DataWarehousing-].

Embora esta fase exista praticamente em todos os projectos de *data warehousing* não é necessariamente obrigatória. É possível construir os cubos do *data warehouse* a partir da informação operacional. Caso esta seja a solução adoptada, é bem provável que se venham a enfrentar os seguintes problemas e limitações:

- Será mais difícil efectuar o mapeamento entre os campos existentes num esquema de dados normalizado com as dimensões e métricas de um cubo OLAP.
- Será mais difícil efectuar o mapeamento caso os dados tenham de ser obtidos a partir de mais do que uma tabela da base de dados.
- Torna-se praticamente impossível realizar tais operações caso a informação necessite de ser *cleansed*, homogeneizada, ou modificada antes de ser utilizada na construção dos cubos.
- É possível obter conflitos de recursos caso se tente construir os cubos a partir de dados que estejam a ser utilizados por outros processos ao mesmo tempo.

A maioria das organizações armazenam os seus dados em formatos e localizações diferentes, o que pode dificultar o processo de transformação de dados. De modo a efectuar os processos de tomada de decisão, a informação necessita de ser deslocada dos sistemas transaccionais para o *data warehouse*. Durante este processo de “deslocação da informação” ocorrem outros processos, como por exemplo, a transformação dos dados, processos de purificação (*cleansing*). A todo este conjunto de operações sobre a informação chama-se de processo de transformação dos dados, ou de uma forma mais abreviada, ETL. [Larsen00]









4.2. ETL com o Microsoft SQL Server 2000: DTS

No caso de estudo analisado a ferramenta de ETL utilizada foi a ferramenta que acompanha o Microsoft SQL Server 2000 e que se designa de Microsoft SQL Server 2000 DTS (*Data Transformation Services*).


A principal razão para a escolha desta ferramenta recai sobre o facto de ser possível aceder à informação a partir de um conjunto diversificado de fontes de dados e à variedade de tarefas oferecidas no processo de transformação dos dados. Para além destes dois aspectos, a

ferramenta DTS possui uma interface gráfica, que permite reduzir o elevado tempo geralmente dispendido nesta tarefa do ciclo de implementação.

Entre os tipos de fontes de dados passíveis de aceder encontram-se bases de dados SQL Server, ficheiros Microsoft Access, ficheiros de texto, folhas de Microsoft Excel e outros tipos de bases de dados, tais como, dBase 5, Paradox. Quanto às tarefas disponibilizadas pela ferramenta elas encontram-se seguidamente descritas:

- **Transform Data Task** (): trata-se da tarefa básica disponibilizada pelo DTS. Esta tarefa compreende uma origem e uma fonte de dados e tem duas variações básicas. Na primeira, os campos da fonte de dados são directamente copiados para o destino. Na segunda variação são utilizados *scripts* ActiveX de forma a transformar cada registo à medida que vai sendo importado.
- **Data Driven Query Task** (): esta tarefa possui toda a funcionalidade da tarefa Transform Data Task mas oferece a possibilidade de actualizar ou apagar registos de destino com base em cada registo individual de origem. O seu nome advém do facto de os dados de cada registo serem utilizados no preenchimento de parâmetros de *queries*.
- **Bulk Insert Task** (): possibilita o método mais rápido de importar registos para uma base de dados SQL Server a partir de um ficheiro de texto.
- **Copy SQL Server Objects Task** (): esta tarefa fornece a possibilidade de transferir objectos e dados de uma base de dados para outra base de dados SQL Server. Os objectos possíveis de transferir são tabelas, índices, *constraints*, *views*, *stored procedures*, *rules*, *defaults* e *logins*.
- **Execute SQL Task** (): esta tarefa é utilizada para executar um comando SQL numa base de dados utilizada no DTS.
- **ActiveX Script Task** (): esta tarefa permite executar um *script* ActiveX. Os *scripts* podem ser criados a partir do Microsoft Visual Basic Scripting Edition, Microsoft Jscript ou PerlScript. Os *scripts* podem ainda referenciar objectos COM.
- **Execute Process Task** (): utilizada para correr um programa externo ou um ficheiro *batch*.
- **Send Mail Task** (): tarefa utilizada para enviar uma mensagem electrónica através do SQL Mail⁷.

⁷ O SQL Mail consiste num mecanismo que permite enviar e receber mensagens electrónicas geradas pelo Microsoft SQL Server. Na recepção, as mensagens podem consistir no relatório de estado de uma tarefa ou de uma alerta.

- **File Transfer Protocol (FTP) Task** (

Qualquer processo de transformação de dados tem subjacente uma origem e um destino para os dados. Os dados originais sofrem um conjunto diversificado de transformações entre a origem e o destino que corresponde às tarefas acima referidas. Ao conjunto das transformações entre origem e destino dos dados, das ligações à fonte e ao destino dos dados,

⁸ Esta tarefa só está disponível caso o servidor Microsoft SQL Server 2000 Analysis Services esteja instalado.

e das regras de precedência entre diferentes tarefas dá-se a designação de *package*. Pode definir-se um *package* como sendo uma unidade de programação executável que corresponde a um tipo de objectos de alto nível no modelo de objectos do DTS. Na Figura 4.1 pode observar-se um *snapshot* da área de trabalho do DTS onde se pode construir um *package*.

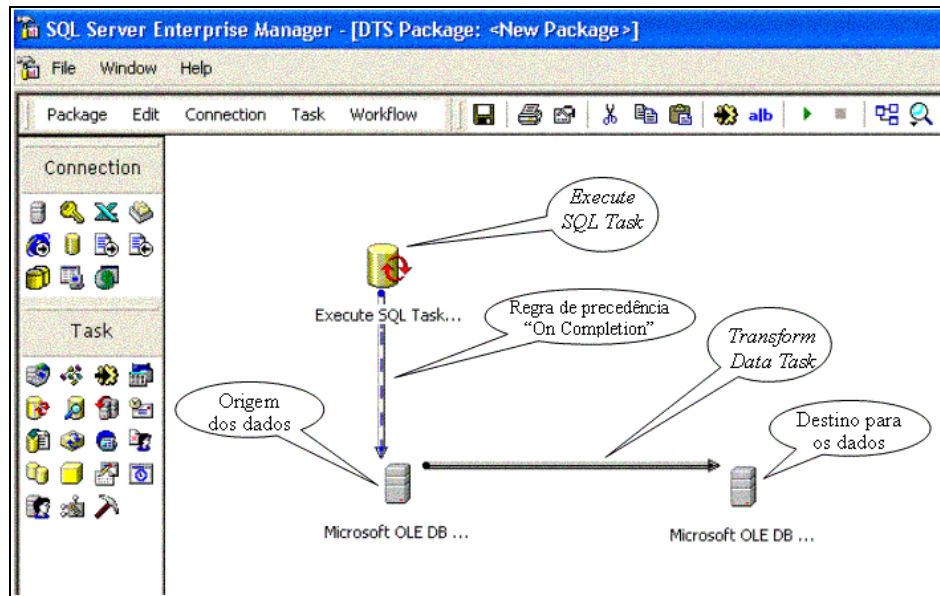


Figura 4.1 – Área de trabalho do DTS que possibilita a construção visual de um *package*

Como pode ser observado neste *package* exemplificativo foram definidas duas ligações de dados (OLE DB), uma tarefa “Execute SQL Task”, uma tarefa “Transform Data Task” entre a origem e o destino dos dados e uma regra de precedência – “On Completion”.

4.3. DTS aplicado ao Caso de Estudo

Uma vez escolhida a ferramenta de ETL, era então altura de definir os *packages* que permitiriam efectuar o processo de transformação de dados. Foi necessário ter em linha de conta que os dados disponíveis para além de se encontrarem segundo uma visão normalizada, residiam no formato DBF, ou seja, exteriormente ao SQL Server. Como tal, houve necessidade de definir duas fases de ETL diferentes, assinaladas no diagrama de componentes da Figura 4.2 por DTS_1 e DTS_2.

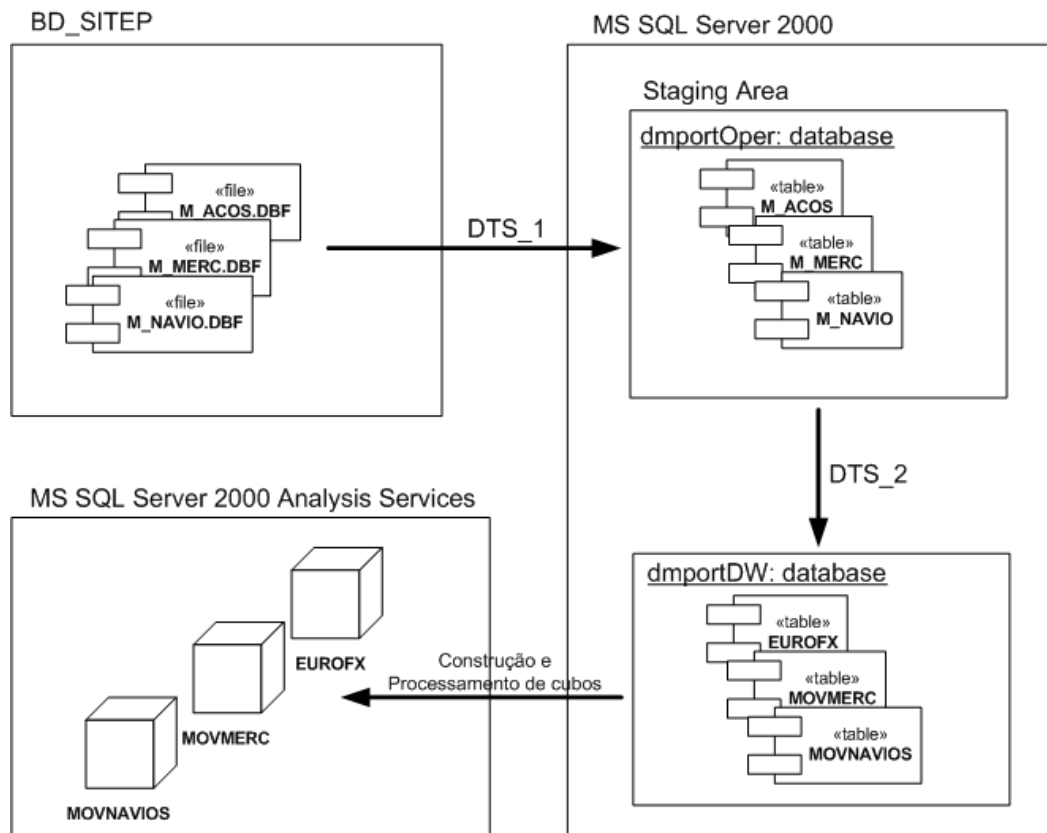


Figura 4.2 – Etapas do processo de transformação de dados

A fase DTS_1 corresponde ao carregamento dos dados para uma base de dados do SQL Server que sirva de área de retenção (*Staging Area*), ou seja, os dados não são alterados – são apenas copiados para uma zona diferente da parte operacional do sistema. Na definição dos *packages* esta fase corresponde ao *package* “Leitura dos ficheiros dBase 5” descrito na Secção 4.3.1. A fase DTS_2 corresponde ao processo de transformação de dados propriamente dito. Compreende os *packages* “Criação e actualização das tabelas de dimensões”, “Processamento dos mov. De navios, mov. De mercadorias, entidades e ocupação de cais” e “Processamento dos quadros para o Eurostat”, descritos nas Secções 4.3.2, 4.3.3 e 4.3.4 respectivamente. Nesta fase os dados são trabalhados entre a base de dados **dmportOper** (visão normalizada) e a base de dados **dmportDW** (visão desnormalizada). Esta fase é responsável pela criação de todas as tabelas de suporte ao *data mart* – tabelas de dimensão e tabelas de factos.

O modelo de dados da base de dados **dmportOper** foi anteriormente apresentado na Figura 3.5. O modelo de dados da base de dados **dmportDW** poderá ser consultado no Apêndice A.2.

No Apêndice A.2 pode observar-se que a base de dados **dimportDW** foi desenhada segundo uma perspectiva multidimensional, embora resida ainda num servidor relacional. Na parte central da figura estão tabelas relacionais que irão alimentar as tabelas de factos, ou seja, os cubos. Na parte esquerda e direita da figura encontram-se as tabelas que irão permitir carregar as tabelas de dimensão. Uma vez que as dimensões são partilhadas por vários cubos foram desenhadas diferentes ligações a partir de cada dimensão para diferentes cubos.

Numa fase posterior do processo ocorre a construção e o processamento dos cubos de suporte ao *data mart*. Uma vez que foi utilizado o tipo de armazenamento de dados MOLAP (Multidimensional OLAP), os dados são transferidos do MS SQL Server para o servidor Analysis Services. Visto os Analysis Services funcionarem como um servidor independente do SQL Server e o tipo de armazenamento de dados utilizado ter sido o MOLAP, não se torna necessário ter o servidor SQL Server activo para se ter acesso aos dados do *data mart*. O contrário não aconteceria caso se optasse pelo tipo de armazenamento ROLAP ou HOLAP.⁹

Tendo em conta os requisitos exigidos pela nova configuração do sistema, enunciados na Secção 3.3, e as etapas DTS_1 e DTS_2, foram definidos os quatro *packages* que se descreve nas seguintes secções.

4.3.1. O *package* “Leitura dos ficheiros dBase 5”

Como já foi referido anteriormente, o sistema SITEP permite a exportação dos dados para ficheiros de formato dBase 5 (extensão DBF). O *package* “Leitura dos ficheiros dBase 5” tem por objectivo efectuar a leitura dos dados provenientes dos ficheiros DBF e transferi-los para uma área temporária, designada por *staging area*.

Tal como em qualquer outro processo DTS, neste *package* foram definidos uma fonte e uma origem de dados. Para fonte de dados definiu-se os ficheiros DBF. E para destino dos dados definiu-se a *staging area* (chamada no protótipo desenvolvido, de base de dados operacional).

Neste *package*, tal como a Figura 4.3 demonstra, definiram-se outras tarefas para além das *Transform Data Tasks*. As outras tarefas definidas, que podem ser observadas do lado direito e esquerdo da Figura 4.3 designam-se de “*Execute SQL Task*” e têm por objectivo efectuar a criação das tabelas referentes aos factos mais importantes do sistema – do lado direito, o

⁹ O leitor pode encontrar uma explicação mais detalhada sobre os diferentes tipos de armazenamento de dados na Secção 2.2

movimento de navios, mercadorias e acostagens, e a criação de algumas tabelas referentes a códigos internos – do lado esquerdo, tipo de navios, tipo de carga, terminais e portos.

Foram ainda definidas regras de precedência neste *package* – uma outra característica do DTS que o torna como uma ferramenta bastante útil nos processos de transformação de dados. No caso deste *package*, a única regra de precedência definida foi a regra designada de “*On Success*”. A regra de precedência “*On Success*” indica que uma determinada tarefa de uma *package* só deve ser iniciada caso a tarefa anterior tenha sido executada com sucesso.

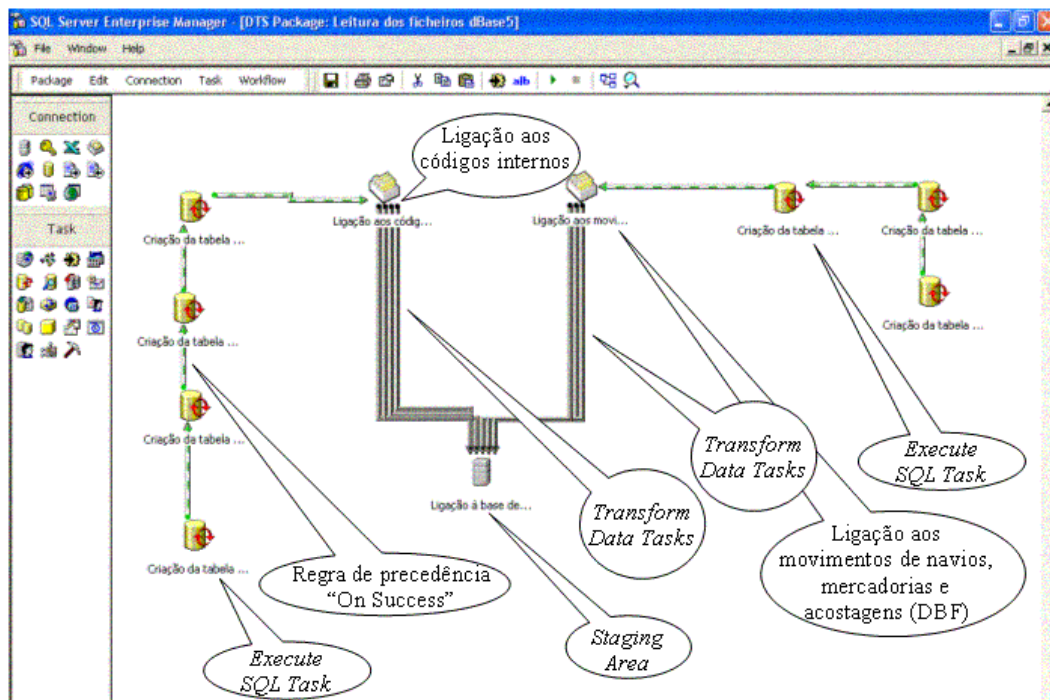


Figura 4.3 – *Package* referente à leitura dos ficheiros DBF

4.3.2. O *package* “Criação e actualização das tabelas de dimensões”

Em todos os projectos de *data warehousing* existe um conjunto de perspectivas, relativas à forma como a organização quer guardar e analisar os dados segundo vários pontos de vista. A essas perspectivas dá-se o nome de dimensões. As **dimensões** são guardadas no *data mart* segundo a forma de tabelas. O *package* “Criação e actualização das tabelas de dimensões” tem por objectivo efectuar a criação das tabelas de dimensões caso estas ainda não existam. Na maioria dos casos esta operação só é realizada uma única vez (para a criação da dimensão), mas, casos hão em que se torna necessário efectuar uma actualização das tabelas de dimensões. Como exemplo temos a dimensão “Tempo”, que na pior das hipóteses

necessitará de ser actualizada uma vez por ano, de forma a incluir o novo ano como membro da dimensão.

Tal como no *package* anterior, para este *package* foram definidos uma fonte e um destino de dados uma vez que também existem *Transform Data Tasks*. Neste caso, a origem dos dados é a base de dados operacional e o destino dos dados é o *data mart*. O *data mart* é uma base de dados segundo o modelo multidimensional e encontra-se mantida pelo servidor SQL Server. Os dados aí guardados serão posteriormente carregados para os cubos da base de dados do servidor Analysis Services.

Existem tarefas definidas neste *package* que criam as tabelas de dimensão e inserem os membros das dimensões através de *scripts* ActiveX. Como exemplo, temos a criação da dimensão referente à Direcção de um navio, ou seja, se é um movimento de Entrada ou um movimento de Saída:

```
if not exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[DIRECCAO]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
BEGIN
CREATE TABLE [dbo].[DIRECCAO] (
[DIRECCAO_ID] [nvarchar] (1) COLLATE Latin1_General_CI_AS NULL ,
[DIRECCAO_NOME] [nvarchar] (7) COLLATE Latin1_General_CI_AS NULL
) ON [PRIMARY]
INSERT INTO DIRECCAO
VALUES ('1','Entrada')
INSERT INTO DIRECCAO
VALUES ('2','Saída')
END
GO
```

Outras tarefas criam apenas as tabelas de dimensões e apoiam-se nas *Transform Data Tasks* para efectuar a cópia dos membros das respectivas dimensões da base de dados operacional para o *data mart*.

Existe uma dimensão que merece uma atenção particular: a dimensão Tempo. A criação desta dimensão passa por duas tarefas do DTS. A primeira, *Execute SQL Task*, tem como objectivo criar a tabela referente à dimensão Tempo caso esta ainda não exista. A segunda, permite construir a dimensão Tempo tendo em conta a hierarquia definida para a dimensão (Ano-Trimestre-Mês). Para realizar a segunda tarefa utilizou-se o tipo de tarefa *ActiveX Script Task* que utiliza a linguagem VB Script na construção do seguinte *script*:

```

\*****
\ Visual Basic ActiveX Script
\ Actualiza os registos da tabela referente ao TEMPO
\*****

Function Main()

    dim connStr
    set myDestConn = CreateObject("ADODB.Connection")

    connStr = "Provider=SQLOLEDB.1;Data Source=(local);Initial Catalog=dmpordDW;user id =
\dmporduser';password='dmporduser'; Trusted_Connection=yes"
    myDestConn.Open = connStr

    dim mySQL, tempo_id, anoStr, trimestreStr, mesStr

    mySQL = "DELETE FROM TEMPO"
    myDestConn.Execute mySQL

    tempo_id = 1
    For ano = 1989 To Year(Date)
        For trimestre = 1 To 4
            For mes = 1 To 3
                anoStr = "" & ano & ""
                trimestreStr = "Q" & trimestre & ""
                mesStr = "" & (trimestre-1)*3+mes & ""

                mySQL = "INSERT INTO TEMPO (TEMPO_ID, ANO, TRIMESTRE, MES)
VALUES (" & tempo_id & "," & anoStr & "," & trimestreStr & "," & mesStr & ")"
                myDestConn.Execute mySQL
                tempo_id = tempo_id + 1
            Next
        Next
    Next

    Main = DTSTaskExecResult_Success

End Function

```

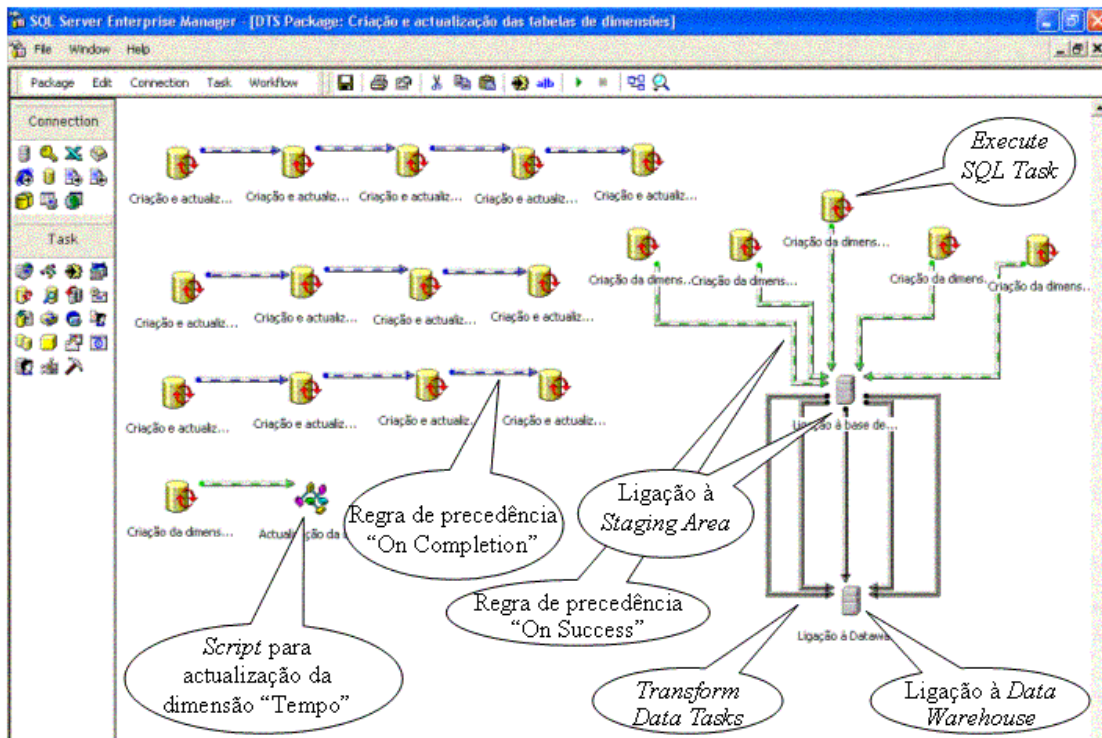


Figura 4.4 – Package referente à criação e actualização das tabelas de dimensões

4.3.3. O *package* “Processamento dos mov. De navios, mov. De mercadorias, entidades e ocupação de cais”

Outro *package* definido na construção do *data mart* é o responsável pelas transformações necessárias na criação das tabelas de factos associadas aos principais aspectos de negócio numa administração portuária, nomeadamente relativos a: (1) movimento de navios; (2) movimento de mercadorias; e (3) indicadores de gestão (em particular, relativos à análise de entidades e de ocupação de cais). No que diz respeito à análise de entidades pretende-se obter informação que referencie quais as entidades responsáveis pelo fretamento de navios, quais os operadores de navegação, etc., ligados ao movimento de navios e ou de mercadorias. No que toca ao indicador de ocupação de cais, pretende-se obter informação referente ao tempo que cada navio se encontra atracado em cada cais, de forma a analisar o rendimento e a taxa de ocupação dos diferentes cais num determinado período de tempo.

Na definição deste *package* foram utilizados dois tipos de tarefas distintas – *Execute SQL Task* e *Data Driven Query Task*. Com a primeira tarefa criam-se as tabelas de factos caso estas ainda não existam. Através da segunda tarefa preenchem-se as tabelas de factos com os dados correspondentes. A tarefa *Data Driven Query* tem como fonte de dados a base de dados operacional e como destino o *data mart*. (O preenchimento das tabelas de factos tem como suporte os dados carregados previamente na *staging area* a partir dos ficheiros DBF.) Através da Figura 4.5 é possível ter uma ideia geral da construção deste *package*.

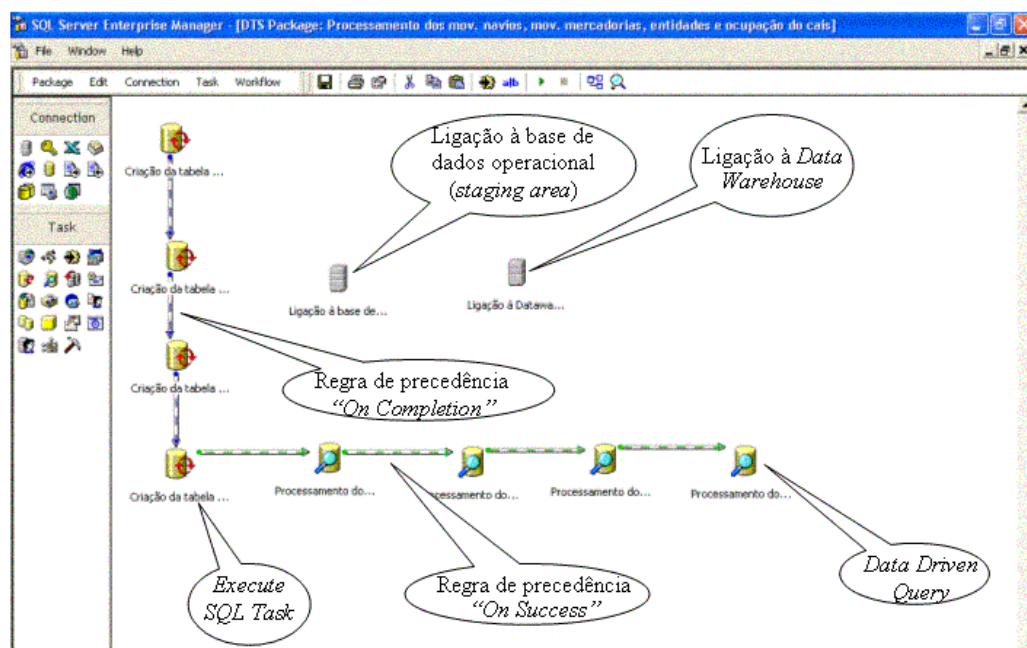


Figura 4.5 – *Package* referente ao processamento dos movimentos de navios, movimento de mercadorias, entidades e ocupação de cais

4.3.4. O *package* “Processamento dos quadros para o Eurostat”

Tal como referido na Secção 3.4, existe um conjunto de quadros que devem ser produzidos com vista ao envio de informação estatística para o Instituto Eurostat da União Europeia. Este *package* tem a responsabilidade de criar a informação que alimentará os cubos correspondentes aos “quadros do Eurostat”.

Este *package* (ver Figura 4.6) foi definido de forma muito semelhante ao *package* anterior. Tem como origem de dados a base de dados operacional (ou seja, a *staging area*) e como destino o *data mart*. Tal como no *package* anterior, utilizaram-se dois tipos de tarefas – a *Execute SQL Task* na criação das tabelas e a *Data Driven Query Task* na transformação dos dados.

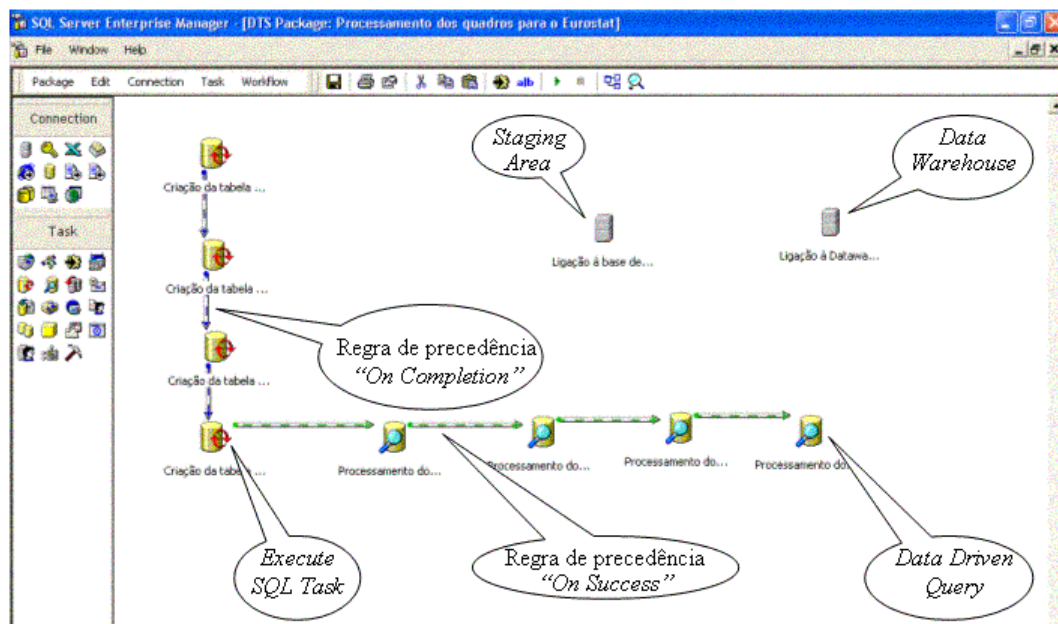


Figura 4.6 – *Package* referente ao processamento dos quadros para o Eurostat

De referir apenas mais um pormenor relevante. Decidiu-se que os cubos na base de dados do Analysis Services deveriam armazenar o maior volume possível de informação. Como tal, cada um destes cubos armazena informação correlacionada que permitirá a produção de mais do que um quadro do EUROSTAT.

Como exemplo, temos o cubo com a informação de suporte aos quadros EuroF1, EuroF2, EuroF3 e EuroF4. A este cubo demos o nome de “EuroFx”. Posteriormente, no Analysis Services foram criados cubos virtuais, que consistem em partes destes mesmos cubos, ou seja, sub-cubos. A Figura 4.7 pretende ilustrar a relação entre o cubo EuroFx e o cubo virtual EuroF1.

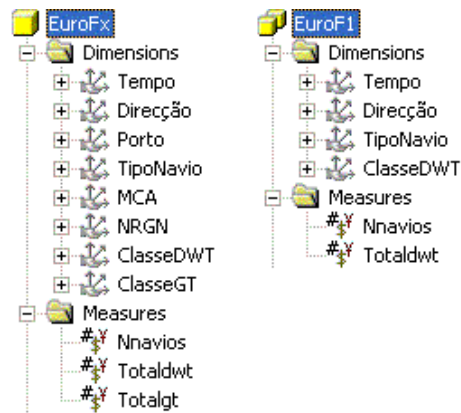


Figura 4.7 – Relação entre o cubo EuroFx e o cubo virtual EuroF1

4.4. Dificuldades, Problemas e Soluções

Ao longo do desenvolvimento deste protótipo, nomeadamente na fase de transformação de dados, foram surgindo alguns problemas os quais foram sendo contornados com maior ou menor dificuldade. Nesta secção serão explicadas as adversidades surgidas e a forma como estas foram sendo solucionadas.

Tipo de Movimento de Navios: Entradas/Saídas (E/S)

No sistema em causa todos os dados referentes a movimentos de navios são registados. Uma informação bastante importante é a indicação do tipo de movimento do navio, isto é, se é um movimento de entrada ou se é um movimento de saída. Caso o movimento seja de entrada é registado num campo essa informação através do código “E”. Caso o movimento do navio seja de saída é registado nesse mesmo campo o código “S”. Porém, na generalidade dos casos, referem-se no mesmo registo, a movimentos de navios com os dois tipos de movimentos (i.e., entrada e saída) através do código “A” (ambos). A Figura 4.8 representa um *snapshot* da tabela de movimentos de navios onde pode ser observada a situação descrita, no campo TIPO_OPE.

	INE_NAVIO	TIPO_OPE	COD_NAVI	NOPTRMAR	AG_NAV	COD_LNAV	PAIS_OR	PORTO_OR
▶	1414	S	2738	ZZ	9997	99	ZZ	958
	1409	S	2738	ZZ	9997	99	ZZ	958
	1414	E	2738	ZZ	9997	99	ZZ	958
	1409	E	2738	ZZ	9997	99	ZZ	958
	1663	A	2738	ZZ	9997	99	ZZ	958

Figura 4.8 – *Snapshot* da tabela de movimentos de navios

Esta situação origina a um esforço suplementar no processo de transformação de dados uma vez que os registos que contenham o código “A” devem ser desdobrados em dois (um

movimento de entrada e um movimento de saída) na passagem da informação para a *data mart*.

Para resolver este problema foi definida uma variável global no *package* chamada de *TipoOperacaoNavio*. Para referenciar a variável global no DTS utiliza-se a seguinte sintaxe: *DTSGlobalVariables("TipoOperacaoNavio")*. Para aceder ao seu valor basta acrescentar o atributo *value*: *DTSGlobalVariables("TipoOperacaoNavio").Value*. O seguinte excerto de código foi utilizado na solução do problema enunciado.

```
If direccao_id = "A" then
    if DTSGlobalVariables("TipoOperacaoNavio").Value ="1" then
        DTSDestination("DIRECCAO_ID") = "1"
        DTSGlobalVariables("TipoOperacaoNavio").Value="2"
        if jaExisteRegisto(tempo_id, "1", tipoNavio_id, porto_id, MCA_id, NRGN_id,
classeDWT_id, classeGT_id)=1 then
            Main = DTSTransformStat_UpdateQuery + DTSTransformStat_SkipFetch
        else
            Main = DTSTransformStat_InsertQuery + DTSTransformStat_SkipFetch
        end if
    else
        DTSDestination("DIRECCAO_ID") = "2"
        DTSGlobalVariables("TipoOperacaoNavio").Value="1"
        if jaExisteRegisto(tempo_id, "2", tipoNavio_id, porto_id, MCA_id, NRGN_id,
classeDWT_id, classeGT_id)=1 then
            Main = DTSTransformStat_UpdateQuery
        else
            Main = DTSTransformstat_InsertQuery
        end if
    end if
else
    if tipo_ope = "E" then
        DTSDestination("DIRECCAO_ID") = "1"
    else
        DTSDestination("DIRECCAO_ID") = "2"
    end if
    if jaExisteRegisto(tempo_id, direccao_id, tipoNavio_id, porto_id, MCA_id, NRGN_id,
classeDWT_id, classeGT_id)=1 then
        Main = DTSTransformStat_UpdateQuery
    else
        Main = DTSTransformstat_InsertQuery
    end if
end if
```

Podem ocorrer duas situações distintas como pode ser observado através da instrução *If* principal. Ou o campo *direccao_id* do registo possui o código "A" ou possui um dos códigos "E" ou "S".

No primeiro caso é necessário inserir dois registos distintos na tabela de factos da *data mart*: o movimento de entrada e o movimento de saída. Para tal recorre-se ao valor da variável global *DTSGlobalVariables("TipoOperacaoNavio")*. Na primeira vez que é inserido um registo o seu valor é "1", o tipo de operação de navio é de entrada e após a inserção do movimento de entrada o valor da variável global é actualizado para "2". Caso já exista algum registo na tabela de factos com os valores indicados – tipo de navio, porto, zona costeira marítima (MCA), nacionalidade de registo do navio (NRGN), classe de *deadweight* e classe de *grossstone* – é efectuada uma operação de UPDATE. Caso contrário é efectuada uma

operação de INSERT. Como se trata da primeira inserção de registo, ou seja, um movimento de entrada, não se pretende que o DTS avance para o próximo registo da fonte de dados e, conseqüentemente, especifica-se essa condição através da instrução `DTSTransformStat_UpdateQuery + DTSTransformStat_SkipFetch`. Na inserção do movimento de saída já se pretende que o DTS avance para o próximo registo da fonte de dados e por isso não se inclui a instrução `DTSTransformStat_SkipFetch`.

O segundo caso, na parte *else* da instrução *if*, tem um tratamento bastante mais simplificado. Basta verificar se já existe algum registo de movimento de navio na tabela de factos e optar por executar uma instrução de INSERT ou por uma instrução de UPDATE. Neste caso não se torna necessário recorrer a variáveis globais.

“O único lugar onde o sucesso vem antes do trabalho é no dicionário.”

Albert Einstein

Capítulo 5.

Operações de OLAP e de *Reporting*

Sumário

Neste capítulo discutem-se as diferenças entre os termos OLAP e *Reporting*. Seguidamente, será apresentado sumariamente o *software* utilizado ao longo desta tese que serviu de apoio na construção de análises OLAP. Relativamente a operações de *Reporting* é feita ainda uma breve abordagem aos Reporting Services, plataforma de *Reporting* da Microsoft utilizada no desenvolvimento da dissertação. Finalmente são ainda apresentados os métodos que nos levaram à disponibilização das operações de OLAP e de *Reporting*.

5.1. OLAP vs *Reporting*

Por forma a introduzir a diferença entre os conceitos de OLAP e de *Reporting* consideremos um hipotético cenário. Imagine-se, por exemplo, que em certo mês do ano recebíamos em nossas casas uma factura de electricidade de valor, à partida, muito superior ao habitual. A nossa primeira questão seria: “Qual teria sido o valor da minha factura de electricidade no mês passado?”. A resposta passaria por consultar a factura fornecida pela empresa de electricidade. Após análise da factura anterior, a questão seguinte poderia ser então: “Qual foi o valor da factura no mesmo mês do ano passado?”. Para obter resposta face a esta questão seria necessário analisar diversas facturas arquivadas no ano anterior. Enquanto que, para a primeira pergunta, obter-se-ia uma resposta rápida apenas consultando um simples relatório

de consumo de electricidade (em papel), para a segunda questão, seria mais prático e célere o recurso a meios tecnológicos. A técnica que permite realizar este tipo de análises e responder a todas estas questões recorrendo a meios tecnológicos denomina-se OLAP. Este simples exemplo permite-nos concluir que a tecnologia OLAP permite-nos analisar a informação de uma forma mais rápida do que através da consulta individual de relatórios.

Nos inícios dos anos 90 a Essbase (na altura propriedade da Arbor mas actualmente propriedade da Hyperion) contratou o “pai” das bases de dados relacionais, E. F. Codd, para tornar claro este novo tema chamado OLAP. Codd definiu então doze regras, mas as seguintes quatro regras são as que mais diferenciam o OLAP do *reporting* [Howson04]:

- (1) **Multidimensional:** os utilizadores analisam valores numéricos a partir de diferentes dimensões, tais como produto, tempo ou geografia. Um relatório, por outro lado, pode ser apenas unidimensional, tal como uma lista de preços de produtos em determinado período de tempo.
- (2) **Rapidez consistente:** enquanto que os utilizadores navegam por diferentes dimensões e por diferentes níveis na mesma dimensão, o OLAP traduz-se em rapidez no acesso à informação. Se o utilizador optar por realizar uma operação de *drill-down* do nível Ano para o nível Trimestre, um tempo de espera de 24 minutos ou de 24 horas é inaceitável. Os utilizadores de relatórios, obviamente, também não pretendem que o acesso aos relatórios seja lento, mas com certeza, alguns relatórios levarão algum tempo a serem construídos e precisam de ser programados para respectiva entrega (*scheduled*).
- (3) **Diferentes níveis de agregação:** para assegurar tempos de resposta aceitáveis, a informação é pré-agregada segundo perspectivas diferentes. O *reporting*, pelo contrário, pode existir no nível de detalhe mais profundo: em vez de vendas por produto, podemos ter o conjunto dos diferentes itens que compõem um número de encomenda em particular.
- (4) **Cálculos cruzados de dimensões** (*cross-dimensional calculations*): múltiplas dimensões implicam cálculos mais complexos. Em OLAP, podemos pretender obter análises percentuais ou quotas de mercado. Este tipo de análises implica o cálculo de subtotais para as vendas relativas a cada região, distrito ou país. Os utilizadores podem analisar esta percentagem de quota de mercado através de diversas outras dimensões,

tais como, ano actual *versus* ano anterior, ou segundo um grupo particular de produtos. Estes cálculos, geralmente, devem ser realizados segundo uma ordem específica e envolvem números não visíveis para os utilizadores. Os relatórios detalhados, pelo contrário, costumam conter sub totais simples ou cálculos de valores disponíveis no próprio relatório.

As principais diferenças entre o OLAP e o *Reporting* resumem-se na seguinte tabela:

	OLAP	<i>Reporting</i>
Dimensões	Múltiplas	Uma ou duas numa <i>cross-tab</i>
Tempos de Resposta	Operações de <i>drilling</i> em poucos segundos	Desde minutos a dias
Nível de Detalhe	Dados sumariados	Dados em detalhe
Cálculos	Dimensões cruzadas	Coluna A; Coluna B ou totais

Tabela 5.1 – Principais diferenças entre OLAP e *Reporting*

Embora as ferramentas de *reporting* tenham um conjunto de características próprias que as diferenciam das ferramentas de OLAP (como pode ser observado na Tabela 5.1), existem actualmente no mercado algumas ferramentas clássicas de *reporting* que também oferecem *features* de OLAP. A funcionalidade mais comum é a operação de *drill-down*, embora outras possam existir, tal como a operação de *slicing*, vulgar nos relatórios parametrizados. A este tipo de *reporting* é comum a designação de *reporting* interactivo. Como exemplo deste tipo de *reporting*, temos os Reporting Services utilizados nesta investigação.

5.2. Enterprise Blocks

Um dos primeiros passos na abordagem ao problema da disponibilização de operações OLAP, consistiu no estudo de algumas ferramentas comerciais de *software* para OLAP. Das várias ferramentas disponíveis no mercado e analisadas optou-se pela utilização do Enterprise Blocks. De seguida, é apresentada a ferramenta e são enunciadas as razões que motivaram esta escolha.

O *software* Enterprise Blocks é constituído por um conjunto de componentes e *web services* que possibilitam o desenvolvimento de aplicações *business intelligence* (BI).

A camada de serviços (*service tier*) é a camada nuclear da arquitectura do Enterprise Blocks. Cada serviço disponibilizado pelo Enterprise Blocks situa-se entre o cliente, que consome os dados, e o fornecedor (*service provider*), carregado pelo serviço, que disponibiliza os dados.

Os serviços do Enterprise Blocks são API abertas. Cada serviço possui um conjunto de métodos que suporta um certo número de funções. Por exemplo, o serviço “Catalog” devolve informação acerca do repositório de metadados que está a ser utilizado. As chamadas aos serviços do Enterprise Blocks podem ser feitas a partir de qualquer ambiente que consuma *web services* XML incluindo Java/.NET/COM a correr numa plataforma Windows ou UNIX. A Figura 5.1 ilustra a arquitectura do Enterprise Blocks.

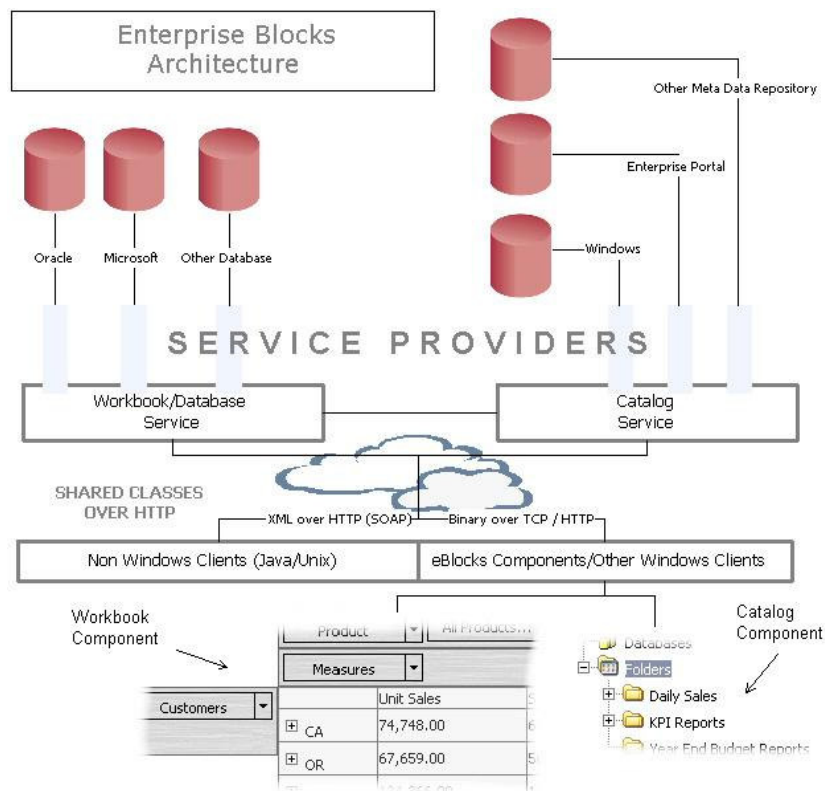


Figura 5.1 – Arquitectura do Enterprise Blocks [adaptado de [Eblocks-]].

Embora os serviços constituam o ponto de entrada e de saída de toda a informação, não comunicam directamente com nenhuma fonte de dados. Os elementos que fazem esta ligação designam-se de fornecedores de serviços (*service providers*) na arquitectura do Enterprise Blocks. Os serviços fornecem apenas uma interface aos consumidores e definem os níveis de segurança no acesso aos fornecedores de serviços. Os serviços encarregam-se de carregar os fornecedores de serviços que interagem directamente com uma fonte de dados específica de modo a retornar a informação necessária.

Existem três tipos de serviços na arquitectura do Enterprise Blocks:

Web Service	Descrição
Catalog	Utilizado para organizar e fornecer o acesso a informação armazenada num repositório de metadados
Database	Utilizado no registo e no acesso a informação existente numa fonte de dados
Workbook	Utilizado em operações de análise a informação proveniente de uma fonte de dados registada

Tabela 5.2 – Tipos de serviços disponibilizados na arquitectura do Enterprise Blocks

Porquê o Enterprise Blocks?

Esta ferramenta de análise conseguiu reunir todas as características necessárias exigidas à partida, e que passarei de seguida a enunciar.

Em primeiro lugar, o Enterprise Blocks possui uma arquitectura aberta, não exigindo por isso, nenhuma plataforma específica nem IDE proprietário. É possível desenvolver as aplicações em Microsoft .Net (utilizando tanto a linguagem de programação C# como o Visual Basic .Net) ou em Java.

Em segundo lugar o Enterprise Blocks permite a ligação a diversas fontes de dados, entre as quais o Microsoft SQL Server 2000 Analysis Services.

Adicionalmente, é possível construir análises OLAP totalmente “*on the fly*”, ou seja, as análises são construídas em *runtime* programaticamente. Existem dois métodos essenciais de construção das análises OLAP no Enterprise Blocks. O primeiro permite especificar a análise pretendida através de uma *query* MDX. O segundo permite ao programador adicionar à análise, dimensão a dimensão, e especificar quais os membros de cada dimensão que devem surgir na análise pretendida.

Relativamente à visualização de análises no formato gráfico este *software* disponibiliza cerca de trinta formatos diferentes de gráficos permitindo ao utilizador ter diferentes pontos de vista relativamente à informação.

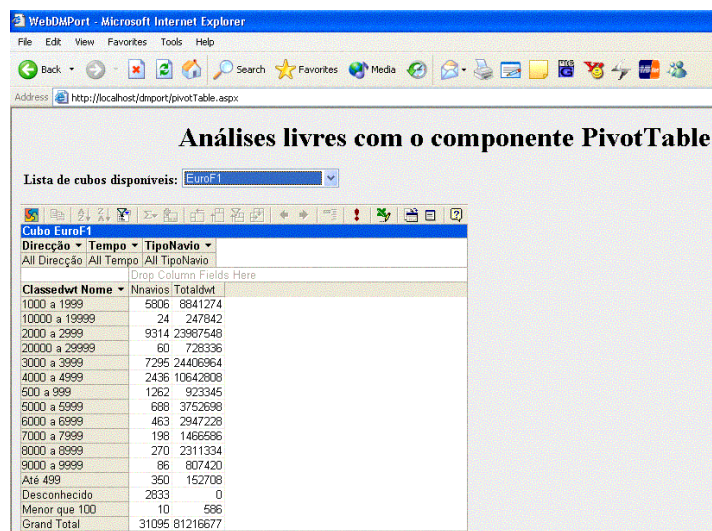
Por último, no que diz respeito à exportação de análises este *software* mostrou também ser bastante poderoso. É possível fazer a exportação das análises segundo o formato tabular e segundo o formato gráfico. Quanto ao formato tabular podemos exportar a análise para o Microsoft Excel, Microsoft PowerPoint, Acrobat Reader (PDF), etc. Quanto ao formato

gráfico a exportação é realizada transformando o gráfico numa imagem que pode ter diversas extensões, como por exemplo, Bmp, Giff, Jpeg, etc.

5.3. Operações de OLAP no DMPort

Construir uma aplicação que suporte operações OLAP implica a obtenção de dados armazenados num *data mart* e a disponibilização de um conjunto diversificado de operações. Nesta investigação foi desenvolvido um protótipo com vista a satisfazer ambos os problemas. Neste protótipo foram disponibilizadas as operações consideradas mais importantes de uma aplicação que suporte operações OLAP. Como tal, foi desenvolvido o código necessário, com a ajuda de *software* já existente, para a realização das seguintes operações de OLAP: *drill down*, *drill up*, *slicing*, *dicing* e *pivoting*. (Ver Secção 2.1.4. para mais detalhes sobre operações OLAP)

O desenvolvimento do código necessário à realização destas operações foi encarado segundo duas formas. A primeira passou pela utilização de um componente da Microsoft e que acompanha o Microsoft Office, o PivotTable. A segunda passou pela criação de uma interface gráfica capaz de disponibilizar a informação relativa aos cubos, dimensões, etc., existente no *data mart*, e consequente programação capaz de disponibilizar essa mesma informação. Neste último caso, podemos optar por apenas visualizar a informação de uma forma tabular ou gráfica sem disponibilizar as operações OLAP, ou ainda, através do *software* Enterprise Blocks, desenvolvido para o efeito, permitir a navegação (entenda-se a realização de operações OLAP) dentro da informação.



Direcção	Tempo	TipoNavio	
All Direcção	All Tempo	All TipoNavio	
[Drop Column Fields Here]			
ClassNavio	Nome	Ninavios	TotalNwt
1000 a 1999		5806	8841274
10000 a 19999		24	247842
2000 a 2999		9314	23967548
20000 a 29999		60	728336
3000 a 3999		7295	24406864
4000 a 4999		2436	10642808
500 a 999		1262	923345
5000 a 5999		688	3752698
6000 a 6999		463	2947228
7000 a 7999		198	1466586
8000 a 8999		270	2311334
9000 a 9999		86	807420
Até 499		350	152708
Desconhecido		2833	0
Menor que 100		10	586
Grand Total		31095	81218677

Figura 5.2 – Snapshot da utilização do componente PivotTable

A Figura 5.2 ilustra a aplicação do componente PivotTable na visualização de informação armazenada na base de dados multidimensional.

O componente PivotTable faz parte de um conjunto de componentes disponibilizados pelo Microsoft Office, ditos OWC (*Office Web Components*), bastante úteis na criação de aplicações BI (*business intelligence*). Na sua utilização num ambiente Web este componente comporta-se como um controlo *ActiveX*, permitindo desfrutar de toda a funcionalidade existente na outra versão do componente para o Microsoft Excel.

Basicamente este componente é constituído por quatro áreas de dados: a área das colunas, a área das linhas, a área de filtros e a área de dados. A Figura 5.3 ilustra as áreas referidas.

The screenshot shows a web browser window titled "dmport - Análises livres com o componente PivotTable - Microsoft Internet Explorer". The address bar shows "http://localhost/dmport/pivotTable.aspx". The main content area is titled "Análises livres com o componente PivotTable". Below the title, there is a dropdown menu for "Lista de cubos disponíveis:" with "EuroF1" selected. The PivotTable is titled "Cubo EuroF1" and has two filter menus: "Tempo" (set to "All Tempo") and "TipoNavio" (set to "All TipoNavio"). The table has columns for "Direccao" (Entrada, Saída, Grand Total) and "Nome" (Nnavios, Totaldwt). The data is organized into a grid with rows representing time intervals and columns representing directions and names. Annotations with arrows point to different parts of the table: a black arrow points to the filter menus (Área de filtros), a blue arrow points to the column headers (Área das colunas), a green arrow points to the data cells (Área de dados), and a red arrow points to the row headers (Área das linhas). A "Home Page" button is visible at the bottom left of the table area.

Classedwt	Entrada		Saída		Grand Total	
	Nnavios	Totaldwt	Nnavios	Totaldwt	Nnavios	Totaldwt
1000 a 1999	2784	4229313	3022	4611961	5806	8841274
10000 a 19999	12	123921	12	123921	24	247842
2000 a 2999	4466	11501909	4848	12485639	9314	23987548
20000 a 29999	30	364168	30	364168	60	728336
3000 a 3999	3511	11754634	3784	12652330	7295	24406964
4000 a 4999	1176	5141823	1260	5500985	2436	10642808
500 a 999	620	452753	642	470592	1262	923345
5000 a 5999	336	1831925	352	1920773	688	3752698
6000 a 6999	222	1414450	241	1532778	463	2947228
7000 a 7999	99	733293	99	733293	198	1466586
8000 a 8999	135	1155667	135	1155667	270	2311334
9000 a 9999	43	403710	43	403710	86	807420
Até 499	175	76354	175	76354	350	152708
Desconhecido	1405	0	1428	0	2833	0
Menor que 100	4	220	6	366	10	586
Grand Total	15018	39184140	16077	42032537	31095	81216677

Figura 5.3 – Áreas de trabalho do componente PivotTable

As principais funcionalidades deste componente na sua aplicação num ambiente Web são as seguintes:

- Realização das operações de *drill down*, *roll up* de modo a definir a granularidade dos dados consultados.
- Realização de operações de *pivoting* (troca de dimensões entre linhas e colunas)

- Realização de operações de *slicing* (selecção de um e um só membro em uma ou mais dimensões).
- Realização de operações de *dicing* (selecção de um conjunto de membros em uma ou mais dimensões).
- Visualização de totais e sub-totais.
- Colocação dos membros por ordem alfabética (crescente ou decrescente).
- Exportação da análise para uma folha de cálculo (Microsoft Excel).

Para a visualização dos dados utilizando o componente PivotTable no protótipo desenvolvido basta definir dois atributos do controlo: (1) a *Connection String* e (2) o cubo. Seguidamente o componente apresenta toda a informação referente a esse cubo. O seguinte excerto de código apresenta o código referido:

```
...  
objPivot.ConnectionString = connStr;  
objPivot.DataMember = objListOfCubes.options[objListOfCubes.selectedIndex].value  
...
```

Tal como foi referido anteriormente, foi desenvolvido um outro método para disponibilização de análises OLAP sem recurso ao componente PivotTable. A construção de análises OLAP sem recurso ao PivotTable desenrola-se segundo a seguinte sequência de passos:

(1) Selecção do cubo e respectivas dimensões a constar, por parte do utilizador

Neste primeiro passo o utilizador terá a missão de escolher o cubo, e respectivas dimensões, sobre o qual deverão recair as análises pretendidas. Em qualquer análise OLAP é necessário especificar o cubo, a partir do qual pretendemos extrair informação, através de análises OLAP. Como tal, foi desenvolvida ao longo deste protótipo, uma interface gráfica que contém quase todas as características inerentes à informação no *data mart*. A interface gráfica, para além de permitir seleccionar um cubo de análise da base de dados multidimensional, permite ainda ao utilizador navegar ao longo das diferentes dimensões, hierarquias, níveis e membros de uma determinada dimensão do cubo. É possível seleccionar o eixo no qual pretendemos colocar a dimensão de análise, isto é, linhas ou colunas, bem como explicitar dimensões de *slicing*. Naturalmente, não foi dispendido neste protótipo demasiado tempo na construção de uma interface muito bem elaborada com todas as funcionalidades de uma aplicação OLAP, uma vez que não é esse o propósito desta tese. No entanto, foram incorporadas nesta fase as funcionalidades consideradas de maior importância e vitais a uma aplicação desta área. A Figura 5.4 apresenta uma ideia geral da interface gráfica disponibilizada na construção de análises OLAP.

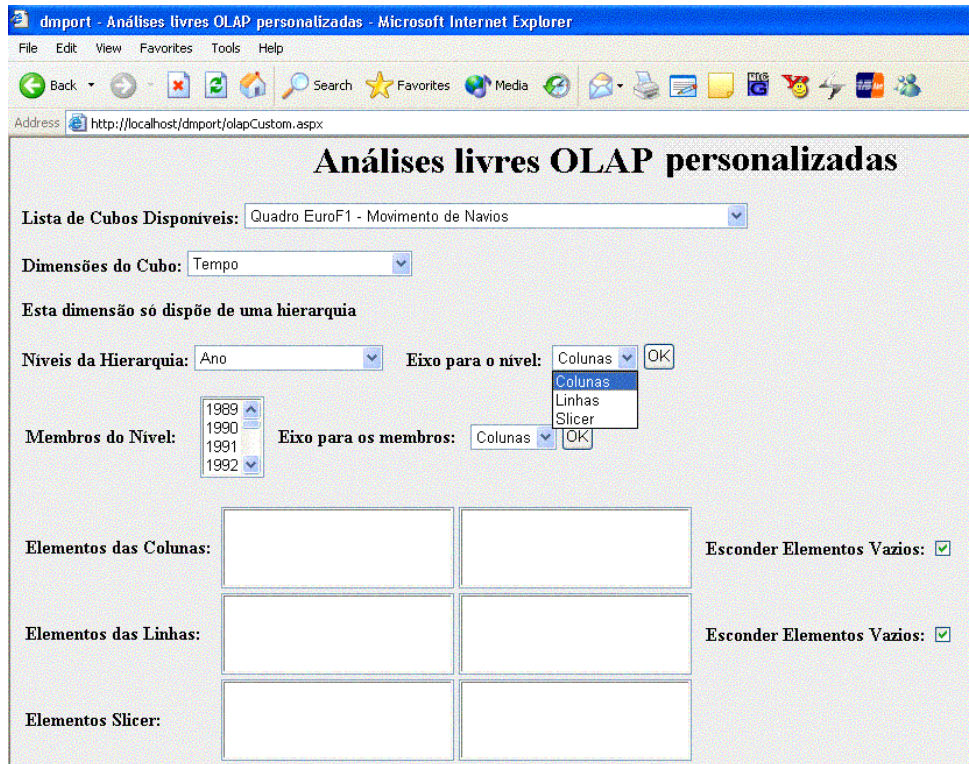


Figura 5.4 – Interface gráfica disponibilizada na construção de análises OLAP

Observando a Figura 5.4 conclui-se que a interface gráfica construída permite ainda indicar se o utilizador pretende ocultar os elementos vazios ou não.

(2) Criação da *query* MDX mediante as dimensões seleccionadas

Este passo implica a construção da *query* MDX responsável pela obtenção de dados a partir da base de dados multidimensional. A construção da *query* MDX baseia-se no cubo e dimensões seleccionadas pelo utilizador no passo anterior. O seguinte excerto de código mostra-nos de uma forma geral como é construída a *query*.

```

Public string createMdxQuery(ListBox listOfColumns, ListBox listOfLines,
                             DropDownList listOfCubes, ListBox listOfSlicers,
                             bool cbColumns, bool cbLines)
{
    string query;

    // construção do SELECT inicial
    query = "SELECT ";

    // construção das colunas
    query += createMainMdx(listOfColumns.Items, 0, cbColumns);
    // construção das linhas
    query += createMainMdx(listOfLines.Items, 1, cbLines);

    // construção do cubo original
    query += " FROM [" + listOfCubes.Items[listOfCubes.SelectedIndex].Value.Split(new
Char[] { '\'}')[0] + "];

    // construção da cláusula WHERE
    query += createMdxSlicer(listOfSlicers.Items);

    return query;
}

```

Basicamente a função responsável pela construção da *query* recebe como parâmetros uma lista de colunas, uma lista de linhas, uma lista de *slicers*, a lista de cubos e duas *check boxes* que indicam se os elementos vazios das linhas e das colunas devem ser visualizados. Cada uma destas listas contém um conjunto de dimensões que devem ser inseridas nas linhas ou colunas da análise consoante a decisão no passo anterior. Na lista de *slicers* constam as dimensões sobre as quais pretendemos efectuar operações de *slicing*¹⁰ (para mais informações sobre a operação de *slicing* consultar a Secção 2.1.4). Como resultado esta função devolve uma *string*, ou seja, a *query* pretendida.

(3) Obtenção do *cellset* a partir da *query* MDX construída no passo (2)

A *query* construída no passo (2) serve neste passo para a obtenção do *cellset* com os valores finais da análise. Na obtenção do *cellset* foi utilizada a classe `Cellset` da biblioteca ADOMD (Microsoft ActiveX Data Objects Multi-Dimensional). O ADOMD é um modelo extremamente hierárquico em virtude dos dados multidimensionais. Existem duas áreas principais de ADO MD: a primeira permite que se navegue na estrutura de um cubo multidimensional e a segunda que se trabalhe com um conjunto de células, que são dados extraídos do cubo. A Figura 5.5 permite ilustrar a hierarquia de objectos do modelo ADOMD.

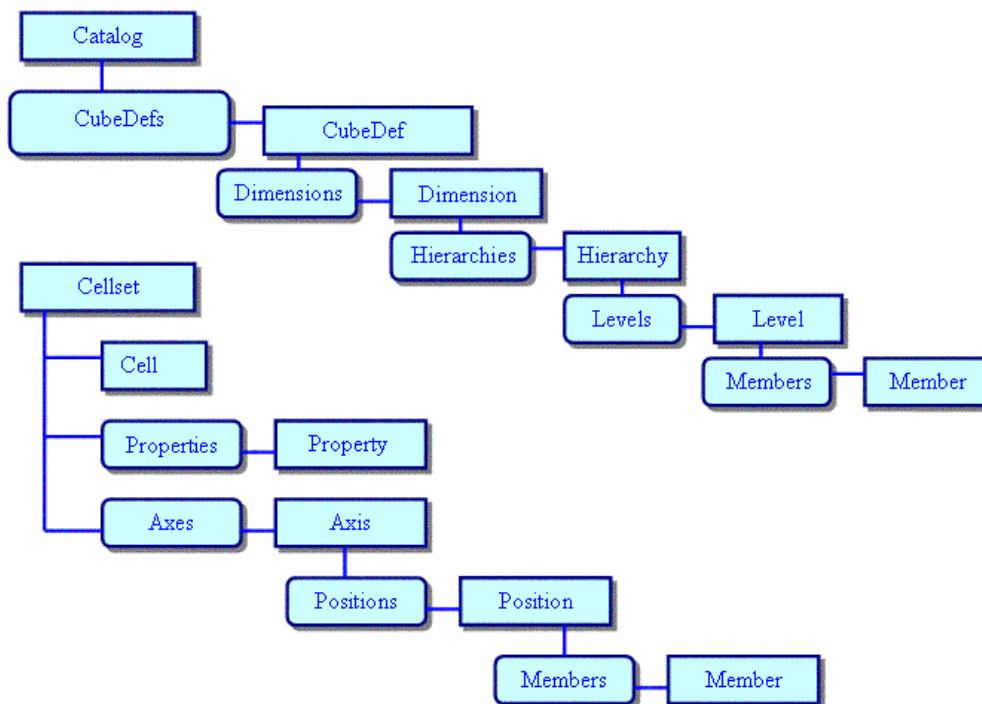


Figura 5.5 – Modelo de objectos do ADOMD [adaptado de [Microsoft-]]

¹⁰ Para mais informações sobre a operação de *slicing* (*slice*) consultar a Secção 2.1.4)

Na extracção do conjunto de dados do cubo foi utilizado o método `Open` da classe `Cellset` que necessita apenas de dois parâmetros: a *query* MDX e uma *Connection String*. Como resultado o método preenche o objecto `Cellset` com os valores finais (como pode ser observado no seguinte código):

```
Cellset cs = new ADOMD.Cellset();  
...  
cs.Open(query, connStr);
```

(4) Construção da matriz com os dados através do *software* Enterprise Blocks

O último passo consiste na construção do resultado visual final da análise pretendida. Neste passo existem duas opções diferentes. Na primeira, o utilizador poderá optar por uma análise dos dados através da construção de uma tabela HTML *standard*. A construção da tabela HTML passará, obviamente, pela análise individual de cada uma das células que constam no *cellset* retornado. Esta opção para além de se revelar um processo moroso, implica que não será possível a realização de operações OLAP após a construção da tabela HTML, nomeadamente, operações de *drill down*. Numa segunda alternativa de visualização da análise é utilizado o *software* Enterprise Blocks, construído precisamente para a área de OLAP como foi referido anteriormente. Nesta alternativa torna-se possível ao utilizador realizar operações de OLAP, como *drill down*, *drill up*, *pivoting*, etc. Em ambas as alternativas é possível visualizar os dados sob o formato gráfico. No primeiro caso utilizando o componente MS Graph da Microsoft, e no segundo utilizando o *software* Enterprise Blocks. As Figuras 5.6 e 5.7 ilustram ambas as alternativas possíveis de visualização das análises.

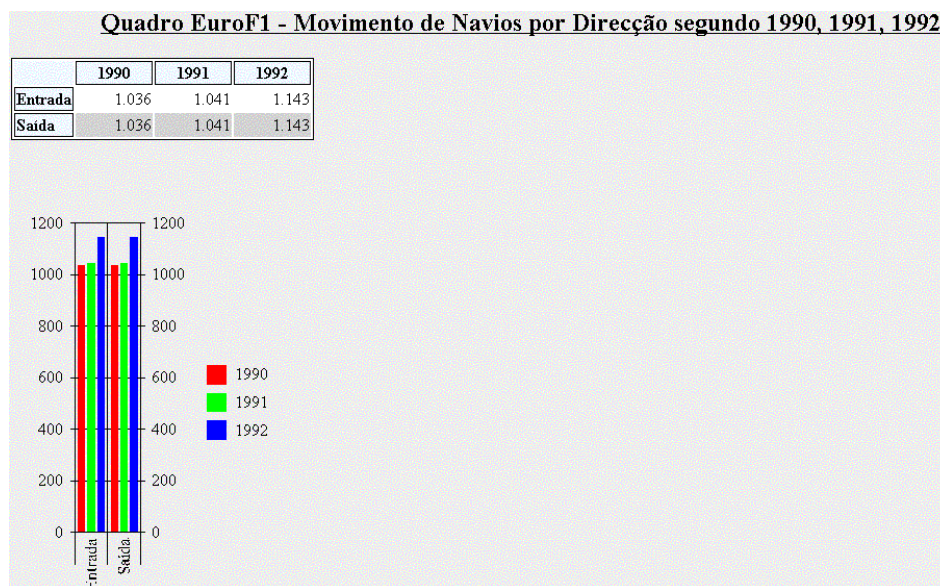


Figura 5.6 – Resultado final de uma análise através de uma tabela HTML

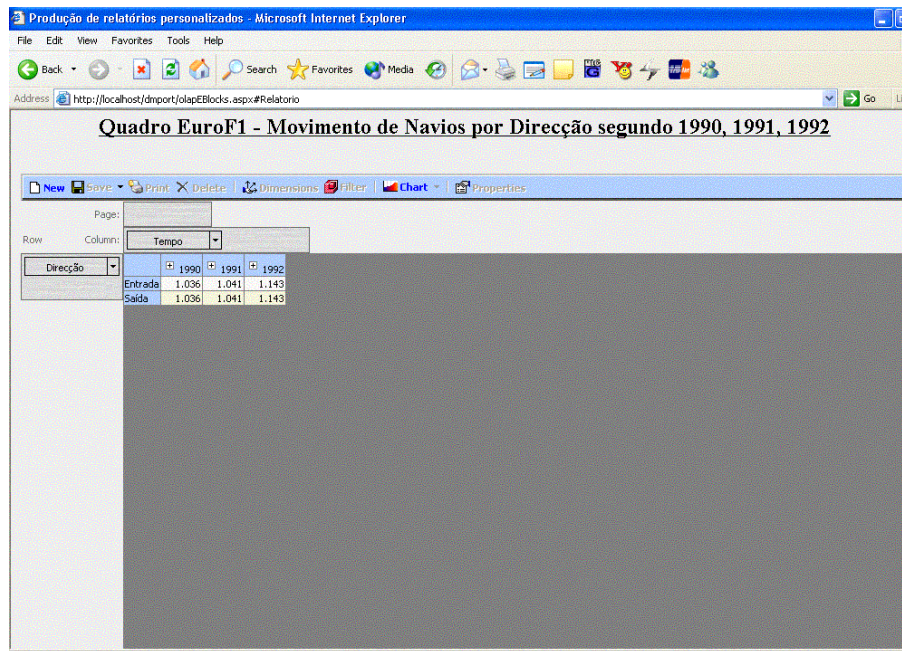


Figura 5.7 – Resultado final de uma análise através do software Enterprise Blocks

A Figura 5.8 apresenta uma análise com duas dimensões no eixo das linhas (Classe de *Deadweight* e Direcção) e uma dimensão no eixo das colunas (Tempo), após a aplicação da operação de *drill down* sobre a dimensão Tempo.

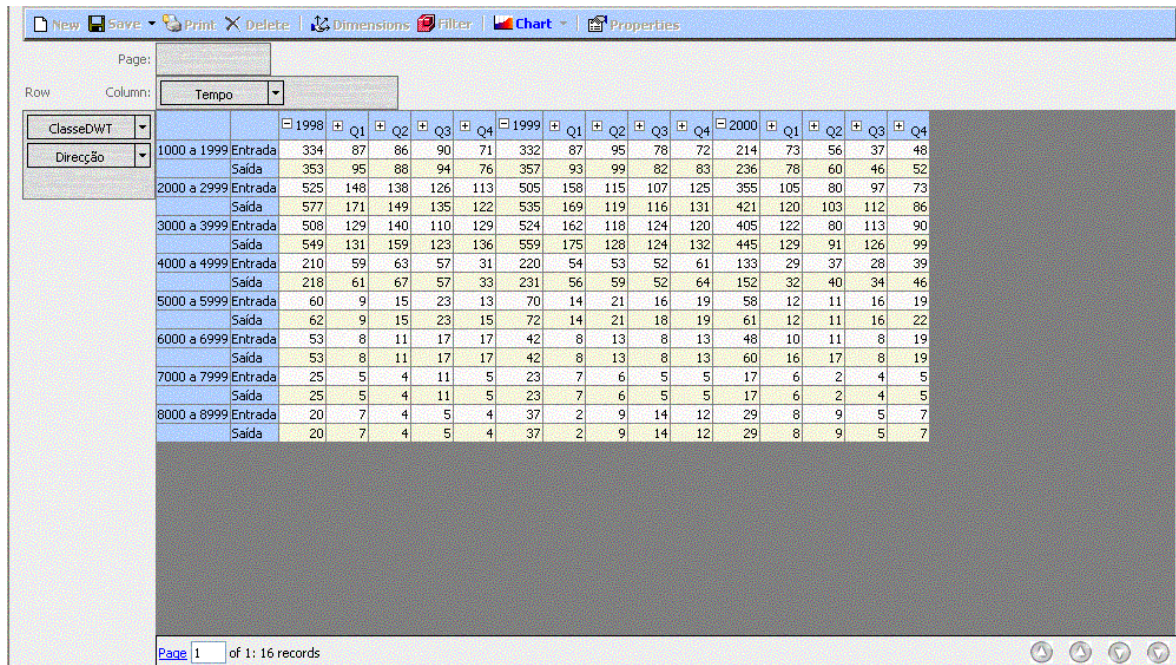


Figura 5.8 – Quadro EuroF1 – Movimento de Navios por Classe de *Deadweight* e Direcção segundo Ano

Por último a Figura 5.9 apresenta uma análise gráfica construída através do software Enterprise Blocks.

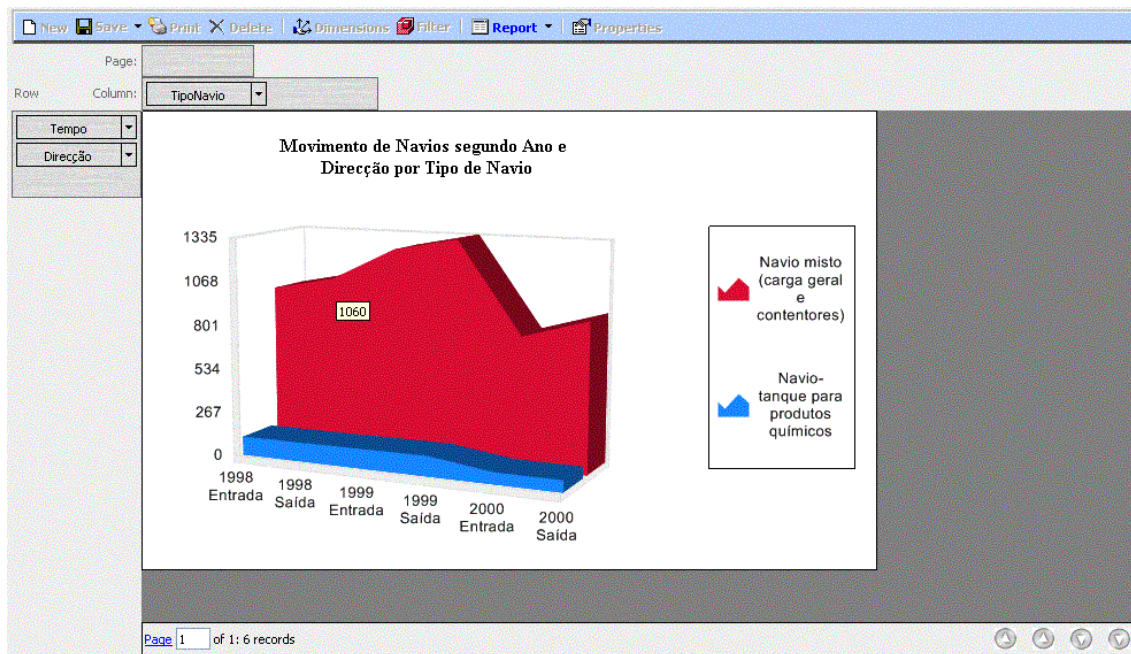


Figura 5.9 – Quadro EuroF1 – Movimento de Navios segundo Ano e Direcção por Tipo de Navio

5.4. Operações de *Reporting* no DMPort

Para a realização das operações de *reporting* no DMPort foi em primeiro lugar analisada a ferramenta Crystal Reports que acompanha o Visual Studio .Net.¹¹ Durante a análise desta ferramenta foram estudadas três formas distintas de poder desenvolver relatórios através do Crystal Reports, que se descrevem em seguida.

5.4.1. *Reporting* com o Crystal Reports

Embora esta ferramenta de *reporting* já tenha sido introduzida brevemente na Secção 2.4 são de seguida descritas, em maior pormenor, as diversas alternativas que o Crystal Reports disponibiliza para a construção de relatórios.

¹¹ Na altura desta análise a ferramenta Crystal Reports era propriedade da Crystal Decisions. Actualmente a Crystal Decisions foi adquirida pela empresa Business Objects.

RDC (Report Designer Component)

O Report Designer Component acompanha o produto Crystal Reports 9¹². O RDC apresenta-se como o melhor método para integrar as capacidades do Crystal Reports numa aplicação.

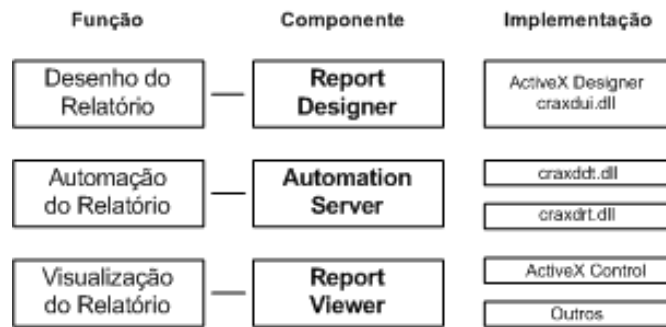


Figura 5.10 – A solução RDC [adaptado de [Oliveira03]]

Como se pode constatar pela Figura 5.10, a solução RDC é composta essencialmente por três componentes: *Report Designer*, *Automation Server* e *Report Viewer*. O *Report Designer* é um componente COM sob a forma de um *ActiveX designer* que permite a criação, a visualização e a modificação de qualquer relatório, seja externo ou interno à aplicação, a partir do próprio ambiente de desenvolvimento. O *Automation Server* é outro componente COM que expõe um modelo de objectos bastante rico, através do qual podemos manipular um relatório em *runtime*. O *Report Viewer*, sendo igualmente um componente COM embora sobre a forma de um controlo *ActiveX*, permite a visualização dos relatórios [Oliveira03].

CrystalDecisions.CrystalReports.Engine

Este componente (modelo de objectos COM) consiste no motor de relatórios que acompanha o Visual Studio .Net para a construção de relatórios.

RAS (Report Application Server)

O Report Application Server é um servidor que providencia um conjunto de serviços base do produto Crystal Enterprise. Este servidor permite o processamento, visualização e modificação em *runtime* de ficheiros do tipo Crystal Reports. Possui um motor de processamento de relatórios e é indicado a programadores que pretendam integrar os seus relatórios em aplicações [CrystalDecisions03].¹³

¹² A versão do Crystal Reports adquirida no desenvolvimento desta tese foi a versão Advanced.

¹³ Esta edição do Crystal Enterprise acompanhou a versão do Crystal Reports no momento da sua aquisição.

No entanto, nenhuma destas alternativas veio a ser a solução adoptada pelas razões a seguir indicadas:

- RDC (Report Designer Component) – embora o Crystal Reports Visual Studio .Net tenha dois viewers (o Windows Form Viewer e o Web Form Viewer) nenhum dos dois pode utilizar um relatório construído pelo RDC.
- CrystalDecisions.CrystalReports.Engine – este motor de relatórios que integra as versões Advanced e Developer do Crystal Reports Visual Studio .Net não contempla as funcionalidades de criação de relatórios dinâmicos (*reports on the fly*).
- RAS (Report Application Server) – o RAS embora permita a criação dinâmica de relatórios, não permite o acesso a fontes de dados OLAP. Apenas permite a visualização de relatórios estáticos baseados em dados OLAP.

Após a análise destas três alternativas do Crystal Reports e após alguns contactos com funcionários da Crystal Decisions especializados em *reporting* concluímos que de todo não seria possível a criação de relatórios dinâmicos acedendo a fontes de dados OLAP, em ambiente .Net utilizando o Crystal Reports, pelo que se tornou necessário arranjar uma alternativa. Depois de algum tempo de investigação, surgiram (em finais do mês de Janeiro de 2004) os Reporting Services da Microsoft.

5.4.2. *Reporting* com os Reporting Services

Uma vez que os Reporting Services já foram apresentados na Secção 2.4.1. apresentamos de seguida a forma como esta plataforma de *reporting* foi integrada no protótipo desenvolvido.

O diagrama da Figura 5.11 ilustra a relação entre os diversos componentes dos Reporting Services bem como a interacção de um utilizador com o protótipo desenvolvido na criação de relatórios dinâmicos.

Reporting Services

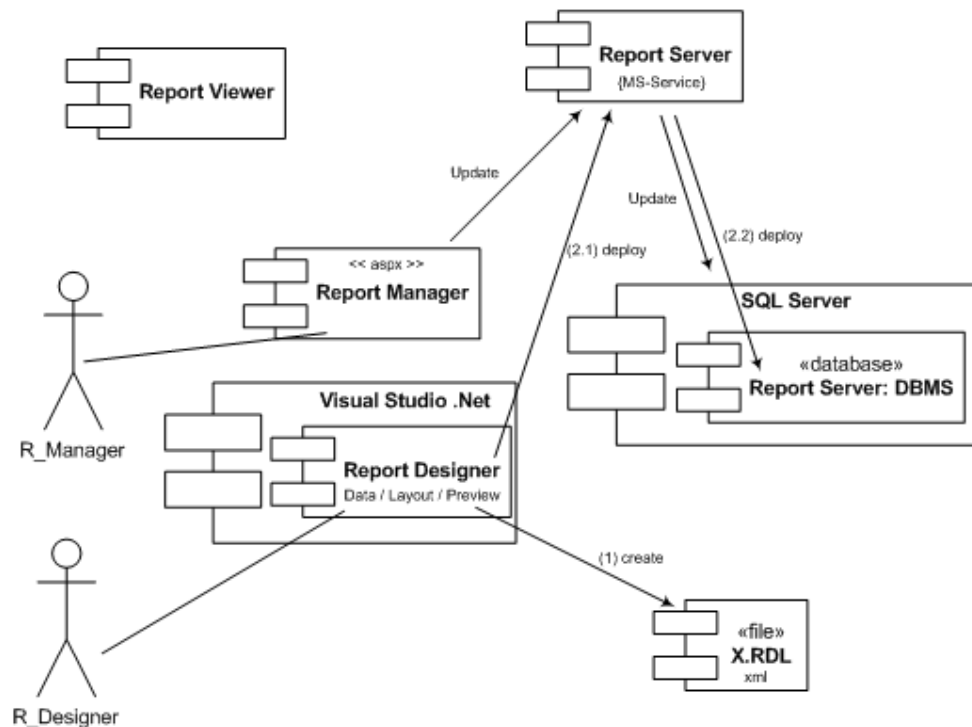


Figura 5.11 – Diagrama de componentes dos Reporting Services

Tal como se disse anteriormente o **Report Server** é o componente principal da arquitectura dos Reporting Services. O **Report Server** é o servidor responsável por todas as interações de um utilizador com qualquer relatório. Sempre que um relatório é criado, modificado ou requisitado, o **Report Server** interage directamente com o catálogo de informação de relatórios (base de dados) existente no SQL Server.

A construção de um relatório através do **Report Designer** resulta posteriormente na publicação (*deploy*) do relatório por parte do **Report Server**, registando a informação necessária na base de dados dos Reporting Services (**Report Server:DBMS** no diagrama de componentes da Figura 5.11 e **ReportServer** por omissão). Este controlo ActiveX contém três tabuladores: **Data**, **Layout** e **Preview**. O tabulador **Data** é responsável pela ligação à fonte de dados e pelas *queries* necessárias no acesso à informação. É importante referir que a arquitectura dos Reporting Services permite a definição de mais do que uma fonte de dados. No tabulador **Layout** o utilizador pode definir a construção visual do relatório, ou seja, se este consistirá numa tabela, numa matriz, num gráfico, ou noutra forma de visualização da informação. Após a selecção de uma destas formas de visualização da informação basta então escolher quais os dados (campos resultantes das *queries*) pretendidos para o relatório. O

visualizador **Preview** torna-se bastante prático uma vez que permite realizar uma visualização do relatório final sem que seja efectuada a publicação (*deploy*) do relatório para a base de dados dos Reporting Services. A Figura 5.12 apresenta um *snapshot* da utilização do **Report Designer** integrado no Visual Studio .Net.

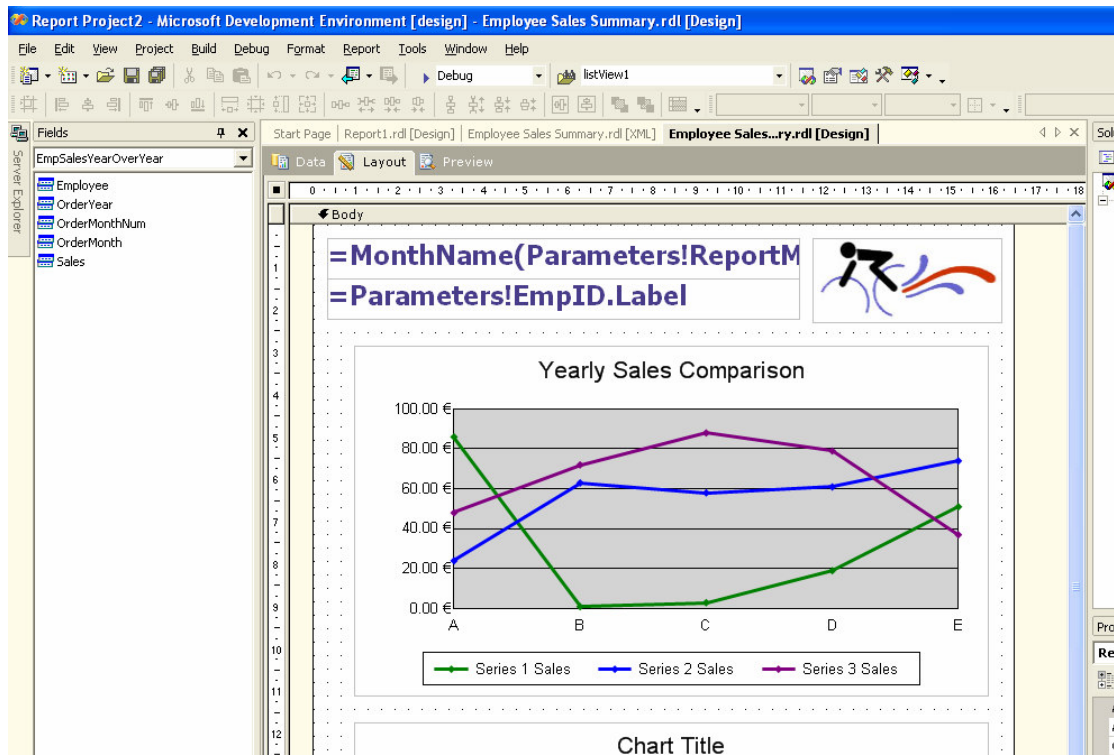


Figura 5.12 - Snapshot da utilização do Report Designer no Visual Studio .Net

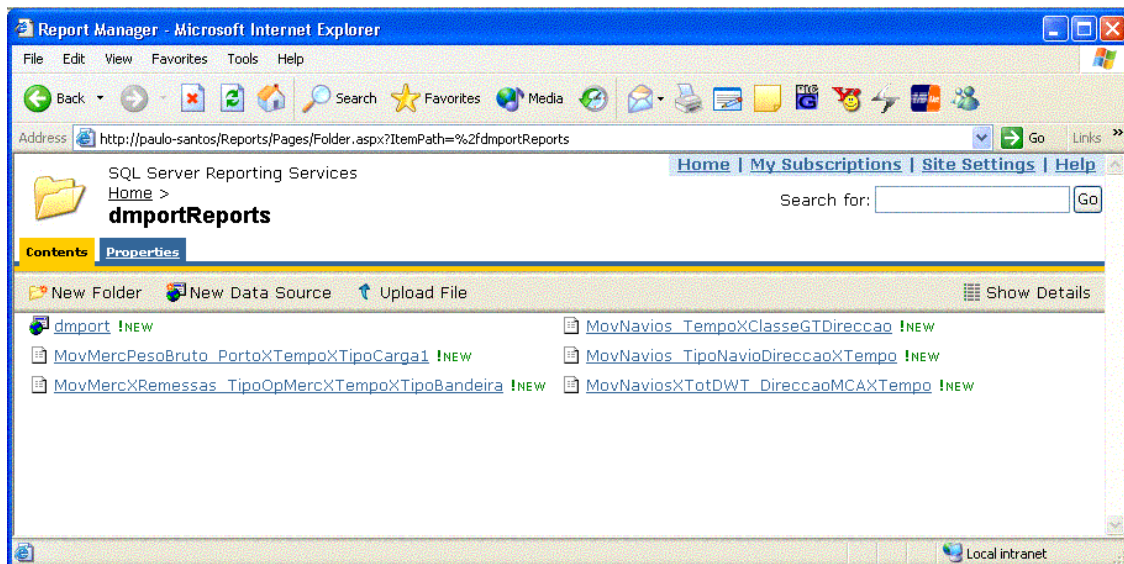


Figura 5.13 – Snapshot da aplicação Report Manager

A aplicação **Report Manager** apenas fornece uma interface gráfica que permite uma interacção mais cómoda com os relatórios, pastas e utilizadores existentes na base de dados dos Reporting Services. Outra forma menos cómoda de realizar tais operações seria através dos métodos disponibilizados pelo Web Service dos Reporting Services (ReportingService.asmx). A Figura 5.13 mostra-nos um *snapshot* da aplicação **Report Manager**, uma vez na pasta *dmpportReports*. Podemos verificar que na pasta *dmpportReports* constam cinco relatórios e uma ligação à fonte dos dados dos relatórios.

Tal como para a realização das operações de OLAP, as operações de *reporting* no DMPORT apoiaram-se na utilização da mesma interface gráfica, embora com algumas nuances ao nível das opções disponíveis na caixa de selecção relativa ao “Eixo”. Como podemos verificar pela Figura 5.14, na caixa de selecção relativa ao eixo existe a possibilidade de indicar como eixo o valor “Parâmetro”. Os Reporting Services permitem-nos criar relatórios parametrizados, ou seja, permitem-nos indicar que determinada dimensão do cubo funcione como um parâmetro disponível no relatório para respectiva selecção de um membro. Daí a existência do valor “Parâmetro” como opção na caixa de selecção relativa ao “Eixo”.

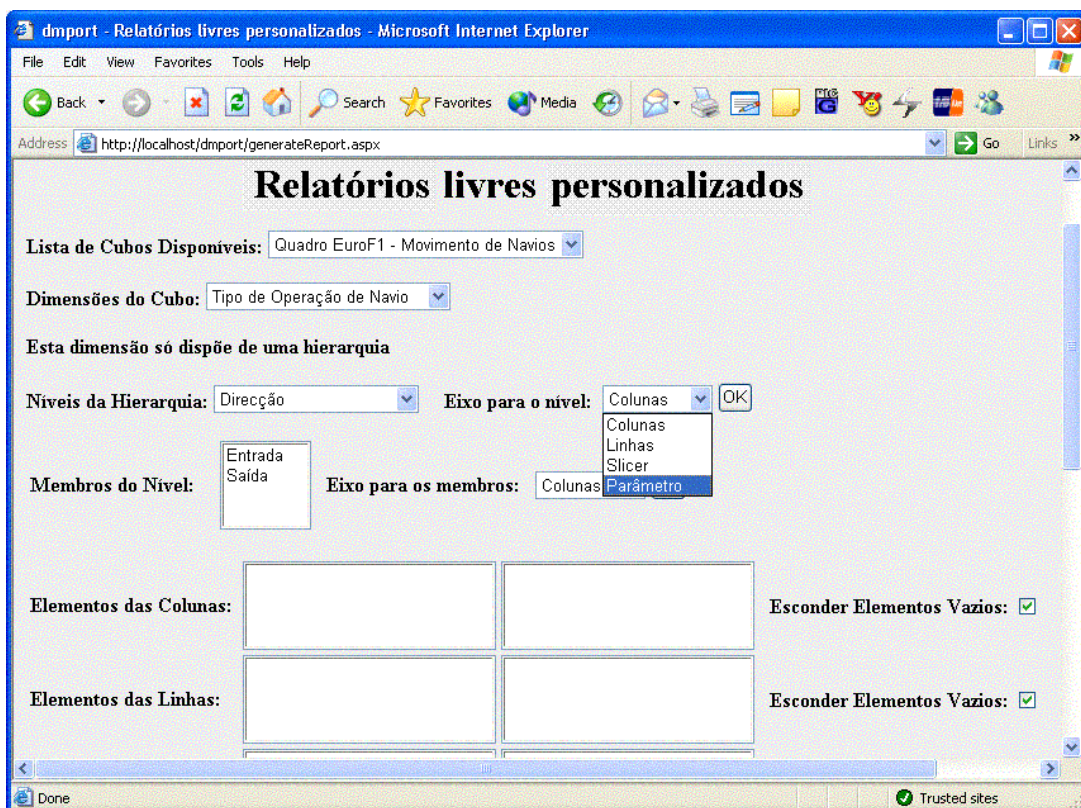


Figura 5.14 – *Snapshot* da interface gráfica para a criação de relatórios personalizados

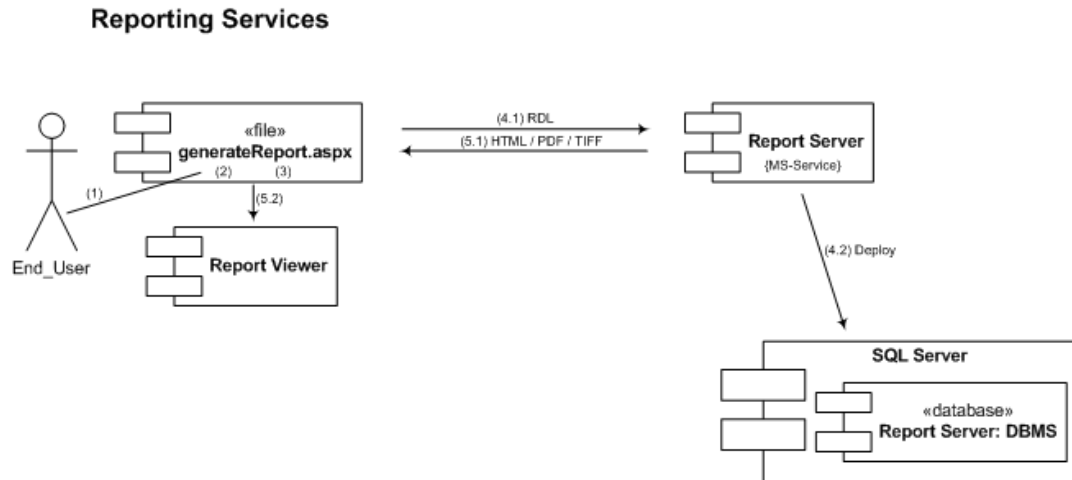


Figura 5.15 – Diagrama de componentes relativo à construção de um relatório personalizado

A construção do relatório personalizado desenrola-se segundo a seguinte sequência de passos (ver acima a Figura 5.15) [Santos05]:

(1) Selecção do cubo e respectivas dimensões a constar no relatório por parte do utilizador utilizando a interface gráfica disponibilizada

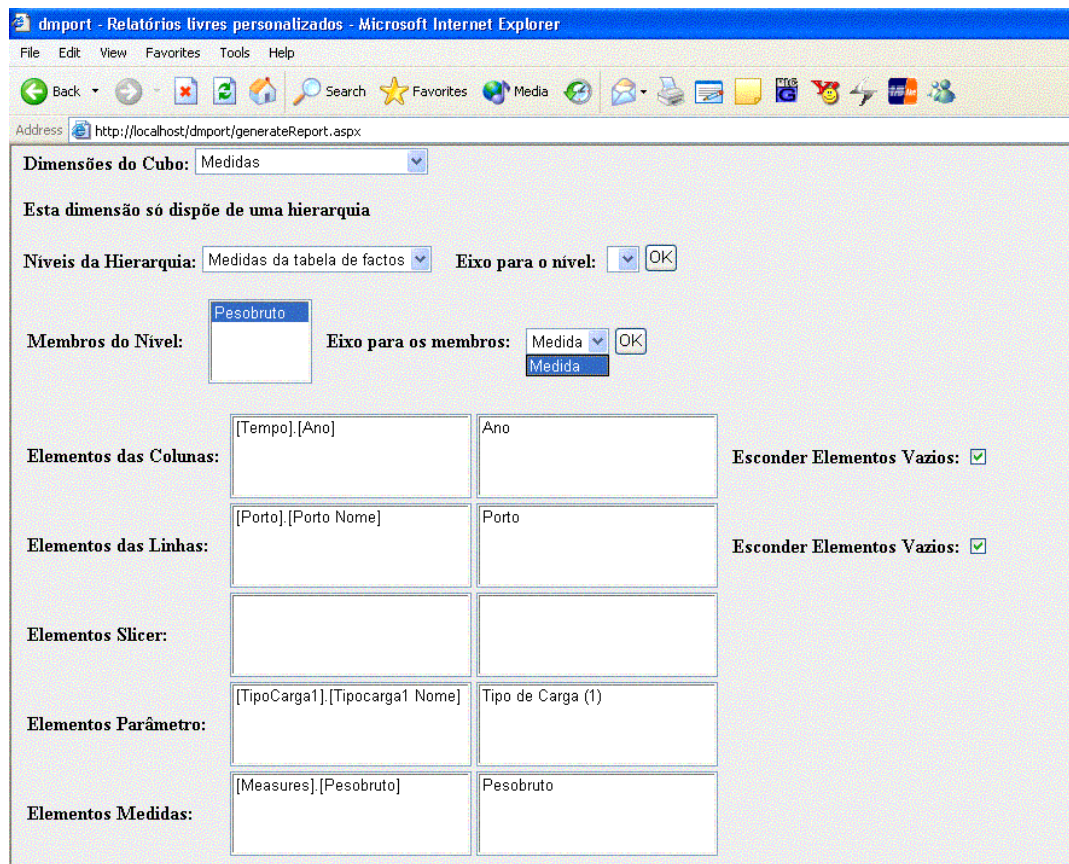


Figura 5.16 – Snapshot da interface gráfica para a criação de relatórios personalizados (2)

Neste passo o utilizador tem a possibilidade de escolher o cubo que deve constar no relatório e quais as respectivas dimensões do cubo, de forma análoga ao descrito na Secção 5.3 para a realização das operações de OLAP. De referir que as medidas do cubo (Measures) são tratadas de uma forma particular. Na construção de um relatório utilizando os Reporting Services é obrigatório colocar as medidas de um cubo na cláusula `Columns` da *query* MDX. Por isso, houve o cuidado de separar as medidas das restantes dimensões como pode ser observado na Figura 5.16, de forma a facilitar a construção da *query* MDX.

(2) Criação da *query* MDX mediante as dimensões seleccionadas

Este passo consiste na geração da *query* MDX de acordo com as dimensões indicadas pelo utilizador. Como exemplo do MDX gerado relativamente às dimensões escolhidas na Figura 5.16 apresentamos o seguinte excerto, fazendo referência à cláusula `Columns` que apenas contém Medidas (Measures).

```
SELECT NON EMPTY {[Measures].[Pesobrutto]} on columns,
NON EMPTY Descendants([Tempo].[All Tempo], [Tempo].[Mes], LEAVES) on rows,
NON EMPTY CROSSJOIN ({[Porto].[Porto Nome].Members},{[TipoCargal].[TipoCargal Nome].Members})
on pages
FROM [EuroA1]
```

(3) Geração do relatório (ficheiro de extensão RDL)

A geração de um relatório segundo os Reporting Services (ficheiro de extensão RDL) consiste na criação de um ficheiro XML com a extensão RDL (Report Definition Language). Uma vez que nesta tese pretende-se a criação dinâmica de relatórios o ficheiro XML teve que ser criado de raiz programaticamente. A geração do relatório implica a criação das fontes de dados, das *queries* MDX e dos objectos de visualização, i.e. tabela e/ou gráfico, de acordo com as opções do utilizador. Uma vez que a especificação RDL cai fora do âmbito desta tese aconselha-se o leitor a consultar [RDLSpec03] caso esteja interessado no assunto.

(4) Publicação do relatório

Este passo consiste na publicação do relatório para o Report Server. A publicação do relatório para o Report Server passa pelo registo de toda a informação associada ao relatório na base de dados dos Reporting Services (**Report Server: DBMS**). Na publicação do relatório foi invocado o método `CreateReport` do Web Service disponível pelos Reporting Services:

```
ReportingService rs = new ReportingService();
rs.Credentials = System.Net.CredentialCache.DefaultCredentials;

Byte[] definition = null;
Warning[] warnings = null;

...
```

```

FileStream stream = File.OpenRead(ut.getXMLValue("reportOutput"));
definition = new Byte[stream.Length];
stream.Read(definition, 0, (int) stream.Length);
stream.Close();

...

warnings = rs.CreateReport(name, "/" + ut.getXMLValue("reportsPath"), true, definition, null);

```

(5) Visualização do relatório publicado através do Report Viewer

Este último passa pela visualização do relatório num ambiente Web. Para tal, foi necessário utilizar o *viewer* desenvolvido pela Microsoft para o efeito, o **Report Viewer**. A visualização do relatório é possível segundo duas formas: a forma em tabela (matriz) e a forma gráfica. As Figuras 5.17 e 5.18 ilustram as duas formas distintas de visualização do relatório. O Report Viewer é um componente bastante poderoso, embora esteja actualmente a ser desenvolvida pela Microsoft outra versão do componente com mais potencialidades. Para além do **Report Viewer** ter capacidades de exportação para diversos formatos de ficheiro (TIFF, PDF, CSV, XML), suporta ainda funções de pesquisa, *zooming* e paginação.

Quadro EuroA1 - Movimento de Mercadorias	1990	1991	1992	1993				199	
	Pesobrut	Pesobrut	Pesobrut	Q1			Q2	Q3	Q4
				1	2	3	Pesobrut	Pesobrut	Pesobrut
Almeria									
Antuérpia (Anvers)				451350					
Anvers (Antuérpia)				451350					
Ashdod									
Aveiro									

Figura 5.17 – Visualização do relatório com o Report Viewer através do formato em tabela (matriz)

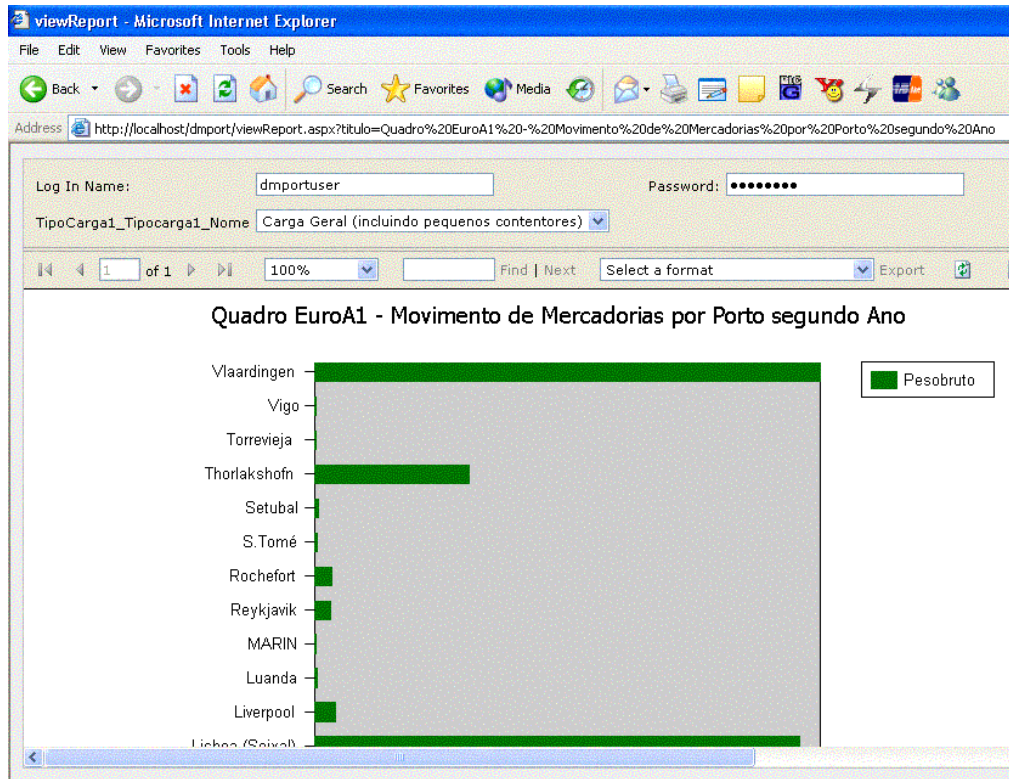


Figura 5.18 – Visualização do relatório sob a forma gráfica com o *Report Viewer*

Com vista à produção de quadros específicos a ser enviados periodicamente ao EUROSTAT (como referido na Secção 3.3), o protótipo desenvolvido possui a capacidade de gravar relatórios. A informação necessária para processar esses relatórios é gravada numa base de dados em SQL Server. Desta forma, é possível definir certos relatórios que são consultados com maior frequência, sem passar pela tarefa morosa da sua construção. Para além disso, a gravação de relatórios tem como vantagem uma maior facilidade na disseminação de informação. Os Reporting Services possuem a capacidade de escalonar os relatórios de modo a serem distribuídos com uma certa periodicidade definida. Aos relatórios gravados pode então ser atribuída a periodicidade de distribuição.

“Se deres um peixe a um homem, ele se alimentará uma só vez. Mas, se o ensinares a pescar ele se alimentará toda a vida.”

Kuan Tsu

Capítulo 6.

Avaliação e Discussão de Resultados

Sumário

Seguidamente desenvolve-se uma avaliação e discussão dos resultados obtidos nesta dissertação, espelhados na sua aplicação ao caso de estudo apresentado no Capítulo 3. São apresentadas algumas considerações gerais sobre os resultados obtidos relativamente a cada uma das fases da metodologia apresentada na Secção 3.3, nomeadamente na adequação da metodologia às necessidades e realidades do domínio da gestão portuária.

6.1. Transformação de Dados

Tal como foi referido na Secção 4.1 a tarefa do processo de transformação de dados, para além de ser uma das tarefas mais importantes de um projecto de *data warehousing*, é aquela que tipicamente requer a maior parte do tempo de desenvolvimento no ciclo de implementação de um projecto desta área. Como tal, torna-se necessário ter um bom conhecimento da área de negócio em causa e dos modelos de dados subjacentes, de modo a facilitar o processo de transformação de dados.

A ferramenta utilizada para o processo de transformação de dados no caso de estudo em mãos mostrou-se ser um bom utensílio de trabalho nesta tarefa da metodologia apresentada na Secção 3.3. Para além de possuir um conjunto diversificado de tarefas relacionadas com a

transformação de dados, os Microsoft SQL Server Data Transformation Services possuem ainda tarefas relacionadas com o processamento de cubos armazenados nos Analysis Services, entre outras tarefas.

Outra das características dos *Data Transformation Services* é a possibilidade de criação de *packages*, ou seja, unidades de programação executáveis. A criação de *packages* permite ao programador, de uma forma simples, encapsular num *package* toda a lógica de transformação de dados referente a um determinado subconjunto da informação em causa, ou mesmo um determinado subconjunto de tarefas que são executadas de forma periódica.

Esta fase do ciclo de implementação revela-se assim de extrema importância e, consoante a sua abordagem, assim dependerá a coerência dos resultados obtidos no final do ciclo de implementação.

Como última observação é importante referir que neste passo da metodologia, para além do programador dever ter um conhecimento profundo dos modelos de dados subjacentes à área de negócio em causa, é bastante importante que os dados de origem tenham qualidade, pois caso contrário, este aspecto negativo reflectir-se-á de modo bastante notório no resultado final.

6.2. Armazenamento de Dados

A decisão de qual o modo de armazenamento de dados é um aspecto primordial em qualquer aplicação desta área. No entanto, essa decisão nem sempre se torna evidente, pois estão em causa diversos factores: o tempo de processamento dos cubos, o tempo de acesso aos dados armazenados, o espaço ocupado em disco pelo cubo, etc. Para que se chegue a uma conclusão final quanto ao melhor modo de armazenamento dos dados é necessário ter em conta todos estes factores, tendo ainda bem presente que não há escolhas ideais. Seja qual for o modo de armazenamento escolhido há que ter em conta que iremos tirar vantagens por um lado, desse modo de armazenamento, mas também, inevitavelmente, sermos confrontados com certos inconvenientes. O que é necessário de um modo geral é que a nossa escolha seja aquela que reúna o maior número de vantagens, consoante a situação.

No caso de estudo em causa estamos perante um volume de dados significativamente reduzido (aproximadamente 2 MB), o que significa que o modo ROLAP seria uma hipótese de escolha viável. No entanto, como foi já referido anteriormente, a decisão sobre o modo de

armazenamento não é trivial e depende acima de tudo dos factores que para o utilizador do *data warehouse* sejam considerados de maior importância. Nos modos MOLAP e HOLAP o tempo de acesso aos dados é bastante mais reduzido que no modo ROLAP. Para além disso, o modo ROLAP implica que haja uma ligação à fonte de dados original (base de dados relacional) sempre que se pretenda realizar operações de OLAP ou de *reporting*, ao contrário do modo de armazenamento MOLAP. Posto isto, o modo ROLAP foi colocado de parte relativamente à sua utilização no caso de estudo.

O modo HOLAP é por vezes a melhor escolha caso não haja espaço suficiente para armazenar grandes quantidades de informação. No acesso às agregações este modo de armazenamento comporta-se de forma semelhante ao modo MOLAP, no entanto o seu desempenho reduz-se de forma substancial, ao nível do modo ROLAP, caso se pretenda aceder aos níveis de informação mais detalhados. Uma vez que neste modo de armazenamento são criadas as agregações na base de dados decisional o tempo de processamento dos cubos é semelhante ao modo MOLAP, diferindo deste apenas no facto dos dados serem deixados na sua origem. Conclui-se por isso que este modo de armazenamento é indicado caso se pretenda utilizar um grande volume de dados e obter um elevado desempenho no acesso às agregações. Por último este modo de armazenamento acarreta ainda a desvantagem da necessidade de existir uma ligação à origem dos dados no acesso aos mesmos.

A escolha do modo de armazenamento para o caso de estudo em questão recaiu sobre o modo MOLAP. Esta escolha foi alcançada devido aos seguintes factores:

- O volume de dados em causa é reduzido (aproximadamente 2MB), pelo que se poderia ter utilizado o modo ROLAP. Contudo, com a utilização do modo MOLAP o espaço de armazenamento necessário para os cubos na base de dados multidimensional não é significativo e a utilização deste modo de armazenamento permite-nos tirar partido de outras vantagens.
- Não se torna necessário haver uma ligação à origem dos dados caso se pretenda realizar acessos. Isto implica que caso a base de dados relacional e a base de dados decisional estejam armazenadas em dois servidores distintos, apenas é necessário que esteja disponível o servidor que contém a base de dados decisional.
- O tempo no acesso aos cubos é bastante mais reduzido mesmo que não tenham sido criadas agregações, uma vez que os dados se encontram segundo um formato multidimensional adequado a acessos a grandes volumes de informação.

- Este modo de armazenamento é provavelmente o melhor modo a menos que haja grandes quantidades de dados que não venham a ser acedidos. Isso não se verifica no caso de estudo em causa uma vez que a maior parte dos cubos construídos constituirão os relatórios a serem enviados periodicamente ao organismo EUROSTAT.

6.3. Operações OLAP

Quando falamos em OLAP falamos na análise a um conjunto de dados históricos, relativamente volumoso, que pretendemos avaliar segundo diversas perspectivas do ponto de vista de uma área de negócio. As diversas perspectivas e os níveis de detalhe pretendidos da informação são acedidos através de um conjunto de operações definidas nesta área, ditas operações OLAP. De entre todas as operações podemos destacar as operações de *drill down*, *drill up*, *slicing*, *dicing* e *pivoting* como sendo as mais usuais e mais importantes, e que estiveram em análise nesta tese.

Ao nível do caso de estudo, demonstrou-se ser possível aplicar de uma forma simples e compreensível o conjunto de operações OLAP ditas de maior importância.

Finalmente, uma última consideração: o conjunto de operações OLAP demonstradas nesta tese pode ser aplicado a qualquer domínio, não estando apenas restrito ao sector da gestão portuária.

6.4. Operações de Reporting

Do ponto de vista dos resultados obtidos quanto às operações de *reporting*, provou-se que estes eram largamente satisfatórios. A utilização da plataforma dos Reporting Services na construção de relatórios personalizados (*on the fly*) contribuiu em muito para que os objectivos propostos fossem alcançados. O facto de a plataforma ser *server-based* facilita a gestão dos relatórios, contribuindo para que haja um repositório de informação comum, centralizado, típico num sistema de *data warehousing*.

Na construção dos relatórios utilizando os Reporting Services alcançou-se, para além da construção do próprio relatório, a possibilidade de realizar operações de *drill down*. Esta operação, embora caia já dentro da área de operações OLAP, só vem enriquecer a capacidade

da plataforma na ajuda preciosa fornecida aos analistas e decisores, na interpretação de dados históricos.

Provou-se com os resultados obtidos que para além dos relatórios apresentarem o aspecto tradicional, no formato de tabela ou matriz, podem ainda apresentar o formato gráfico, que em tudo facilita o trabalho de análise. Recordemos que o formato gráfico para os relatórios confere aos mesmos uma melhoria de análise segundo uma perspectiva qualitativa, ao passo que o formato por tabela, realça a perspectiva quantitativa da informação.

Por último, há a salientar a facilidade de exportação para diversos formatos de ficheiros, existente na plataforma de *reporting* da Microsoft.

“A imaginação é mais importante que o conhecimento.”

Albert Einstein

Capítulo 7.

Conclusões e Trabalho em Aberto

Sumário

Neste capítulo discutem-se as principais conclusões desta dissertação. São ainda definidas futuras linhas de trabalho que possam completar a investigação, tendo por base as contribuições propostas.

Assim, na primeira secção são revistas as questões enunciadas na secção 1.5 que orientaram a investigação e discutem-se as principais conclusões atingidas. Em seguida, na segunda secção, descrevem-se novas direcções de trabalho, com base no trabalho apresentado. Termina-se o capítulo com um comentário final sobre a investigação.

7.1. Conclusões do Trabalho

No início desta dissertação (ver Secção 1.5), foram definidas diversas questões pelas quais esta investigação se orientou e procurámos responder, as quais recordamos nesta secção.

Nomeadamente procurou-se dar resposta às seguintes perguntas:

- Qual é o modelo de dados mais adequado ao processo de tomada de decisão nesta área da gestão portuária? Qual a melhor técnica de armazenamento de informação para a realização do processo da tomada de decisão no contexto da gestão portuária?

- Qual a técnica que se deve utilizar para realizar o processo de transformação de dados entre os dois modelos de dados e repositórios de informação?
- Como disponibilizar um conjunto de operações capazes de auxiliar o processo de tomada de decisão? Como desenvolver um conjunto de operações OLAP de forma a flexibilizar os processos de análise? De que forma é que se pode aumentar a flexibilidade da produção de relatórios já existente no SITEP?

A primeira questão foi abordada no Capítulo 4. Nesse capítulo foram apresentados os principais factores que influenciam na decisão de qual a técnica de armazenamento de dados mais adequada: (1) o tempo de processamento dos cubos; (2) o tempo de acesso aos dados armazenados; e (3) o espaço ocupado em disco pelo cubo. A utilização dos *Analysis Services* facilitou em muito esta parte da investigação, uma vez que este servidor suporta as três técnicas de armazenamento mais conhecidas (MOLAP, ROLAP e HOLAP).

Concluiu-se ainda que não existem escolhas ideais, e que todas as técnicas existentes possuem vantagens e inconvenientes. Para o caso de estudo em questão a escolha recaiu sobre o modo MOLAP embora também pudesse ter sido adoptado o modo ROLAP pelo facto do volume de dados em causa ser bastante reduzido.

A segunda questão foi também abordada no Capítulo 4. Embora a tarefa do processo de transformação de dados exija um elevado esforço e seja uma das tarefas mais morosas, não exigiu neste trabalho um grande esforço de investigação. Foi sim necessário, um estudo aprofundado da área de negócio em causa e dos modelos de dados subjacentes. A tecnologia utilizada, os *Data Transformation Services* da Microsoft, facilitaram em grande parte o esforço necessário nesta tarefa.

No Capítulo 5 (o mais importante desta dissertação) foram apresentadas técnicas capazes de realizar um conjunto de operações de OLAP e de *reporting* que suportam o processo de tomada de decisão na área da gestão portuária. Foram definidas ainda, e apresentadas em concreto, metodologias para a produção dessas mesmas análises aos dados.

Na disponibilização de operações de OLAP foram utilizados o componente *PivotTable* e um software proprietário, o *Enterprise Blocks*. Estes dois métodos distintos permitem flexibilizar de forma significativa os processos de análise comparativamente ao sistema legado existente anteriormente.

Relativamente à disponibilização de operações de *reporting* foi apresentado um método de produção dinâmica e interactiva de relatórios, que permite ao utilizador, em *runtime* e segundo um ambiente *Web* efectuar a selecção de dados pretendida (isto é, realizar um conjunto de *queries* à base de dados de forma transparente). Esta facilidade foi atingida, em grande parte, graças à utilização da nova plataforma de *reporting* da Microsoft, os “*Reporting Services*”.

Ao contrário da maioria dos produtos existentes no mercado na área de *reporting* (nesta investigação estudou-se também o *Crystal Reports*), cujos relatórios são definidos segundo um formato proprietário, os *Reporting Services* definem os relatórios segundo um formato aberto. O facto dos relatórios serem definidos à custa de uma linguagem aberta, o XML, permite aos utilizadores definirem novos elementos para os seus relatórios de acordo com as necessidades do negócio. A esta vantagem, tal como já foi frisado anteriormente, acresce o facto de podermos implementar dinamicamente relatórios (em *runtime*), sem estarmos restringidos a um conjunto dos mesmos definidos previamente (em *design-time*).

Por fim, é importante salientar o facto de o processo de tomada de decisão ser aplicável às mais diversas áreas de negócio. As conclusões obtidas nesta dissertação não se confinam à área dos transportes marítimos, especificamente à gestão portuária, são antes pelo contrário generalizáveis. O subconjunto de funcionalidades referentes à produção dinâmica e interactiva de relatórios foi já aplicado ao projecto gestBarragens (Sistema Integrado de Gestão da Informação para o Controlo de Segurança de Barragens), em desenvolvimento pelo Grupo de Sistemas de Informação do INESC-ID [GestBarragens-].

7.2. Trabalho em Aberto

As contribuições desenvolvidas e as conclusões do trabalho realizado não podem ser considerados como uma meta final ou solução para o processo de tomada de decisão, tanto na área da gestão portuária como na generalidade das diferentes áreas em que este processo possa ser aplicado. Representam, acima de tudo, um estudo sólido sobre o assunto em questão e o princípio de um vasto trajecto de investigação.

Por aprofundar, ficaram questões como a produção dinâmica de relatórios ao nível da sua apresentação gráfica (e.g. definição e gestão de *templates* de relatórios). Este trabalho permite

apenas a produção dinâmica de relatórios ao nível da selecção de dados, isto é, a definição de *queries* à base de dados em *runtime* e de forma transparente para o utilizador.

Acredito que o desenvolvimento de uma ferramenta que suporte o conjunto de funcionalidades atingidas nesta dissertação, juntamente com a possibilidade de produção de relatórios de forma interactiva (a nível da sua apresentação gráfica) possa ser facilmente integrada numa plataforma de apoio à decisão.

Outra questão para a qual também considero importante o seu aprofundamento consiste na disseminação selectiva de informação. Seria importante neste tipo de soluções a existência de uma camada intermédia de segurança que permitisse apenas o acesso à informação por parte dos utilizadores que, para tal, tenham os respectivos privilégios e adicionalmente o envio automático de relatórios via *e-mail*. Este aspecto é facilmente atingível através, por exemplo, da plataforma dos Reporting Services, devido ao seu carácter de extensibilidade, quer ao nível da segurança quer ao nível da distribuição e entrega de relatórios.

7.3. Comentário Final

Estando prestes a concluir esta investigação, onde aprendi diversos conceitos e tecnologias, aprez-me dizer que foi uma experiência bastante agradável e positiva. Mais do que o contributo que esta possa oferecer à ciência e às organizações em geral, serviu principalmente como meio de evolução e maturação científica na área da investigação. Serviu também, por outro lado, como objecto de realização pessoal. Tal como diria Fernando Pessoa: “Deus quer, o Homem sonha, a obra nasce”.

Apêndices.

A.1 Lista de cubos construídos

A.1.1 Cubos com informação exigida pelo Eurostat

Movimento de Mercadorias por Tipo de Carga (1)

EuroA1
Tempo
Tipo de Operação de Navio
Porto
MCA
Tipo de Carga (1)
Peso Bruto

Movimento de Mercadorias por Tipo de Carga (2)

EuroA2
Tempo
Tipo de Operação de Navio
Porto
MCA
Tipo de Carga (2)
Peso Bruto

Movimento de Mercadorias por Tipo de Carga e Grupos

EuroB1
Tempo
Tipo de Operação de Navio
Porto
MCA
Tipo de Carga (1)
GMER
Peso Bruto

Movimento de Mercadorias por Tipo de Carga, Grupos e IMDG

EuroB2
Tempo
Tipo de Operação de Navio
Porto
MCA
Tipo de Carga (1)
GMER
IMDG
Peso Bruto

Movimento de Mercadorias em Contentores ou Ro-Ro

EuroC1
Tempo
Tipo de Operação de Navio
Porto
MCA
Tipo de Carga (2)
Peso Bruto
Unidades
Unidades Cheias
Unidades Vazias

Movimento de Mercadorias em Contentores ou Ro-Ro

EuroC2
Tempo
Tipo de Operação de Navio
Porto
MCA
Tipo de Carga (1)
Tara
TEU

Movimento de Passageiros por Porto e Operadores

EuroD1
Tempo
Tipo de Operação de Navio
Porto
MCA
País Operador
NRGN
Nº Passageiros

Movimento de Mercadorias por Tipos, Portos e Operadores

EuroE1
Tempo
Tipo de Operação de Navio
Porto
MCA
Tipo de Carga (1)
País Operador
NRGN
Peso Bruto

Movimento de Navios por Tipo e Classe de Deadweight

EuroF1
Tempo
Tipo de Operação de Navio
Tipo de Navio
Classe de Deadweight
Nº Navios
Total Deadweight

Movimento de Navios por Tipo e Classe de GrossTone

EuroF2
Tempo
Tipo de Operação de Navio
Tipo de Navio
Classe de GrossTone
Nº Navios
Total GrossTone

Movimento de Navios por Tipo e Classe de GrossTone

EuroF3
Tempo
Tipo de Operação de Navio
Porto
MCA
NRGN
Tipo de Navio
Classe de GrossTone
Nº Navios
Total GrossTone

Movimento de Navios por Tipo e Classe de Deadweight

EuroF4
Tempo
Tipo de Operação de Navio
Porto
MCA
NRGN
Tipo de Navio
Classe de Deadweight
Nº Navios
Total Deadweight

A.1.2 Cubos com registos de movimentos

Movimento de Navios

Mov. Navios
Tempo
Classe de GrossTone
Classe de Comprimento
Terminal
Bandeira
Tipo de Carga (2)
Nº Navios

Movimento de Mercadorias

Mov. Mercadorias
Tempo
Tipo de Carga (2)
Terminal
Tipo de Operação de Mercadoria
Bandeira
Porto
Quantidade
Remessa

A.1.3 Cubos com informação sobre indicadores

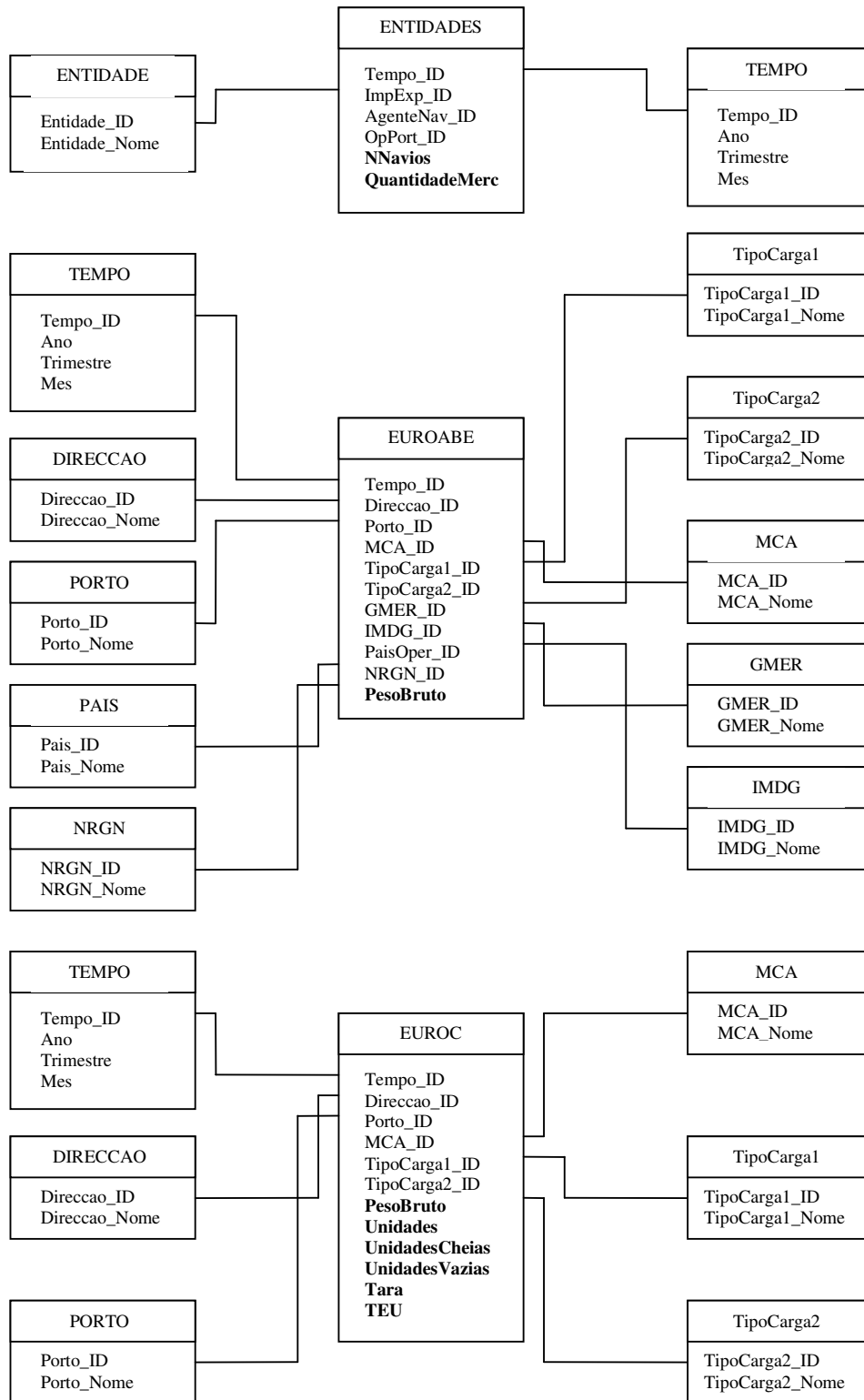
Entidades

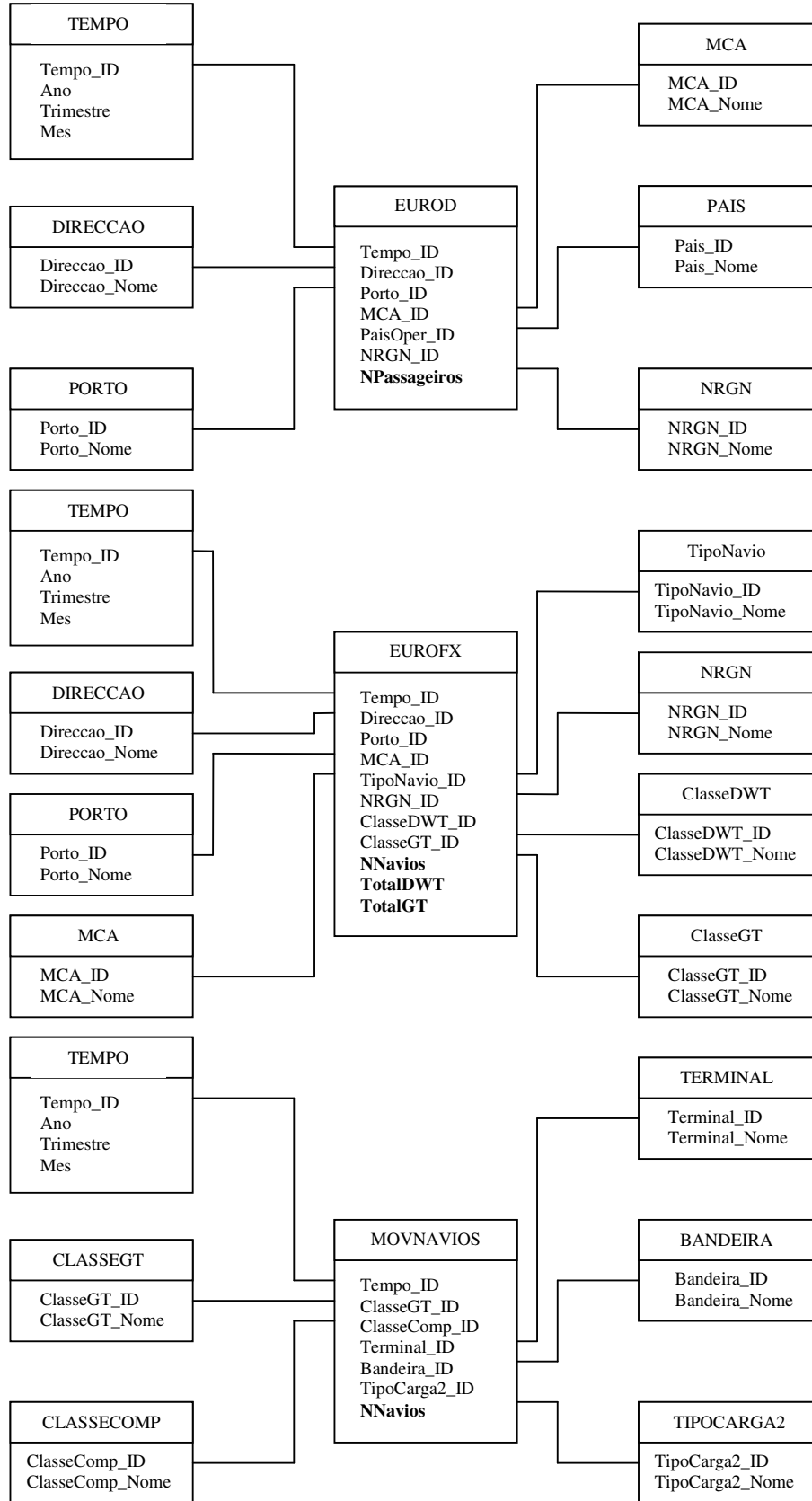
Entidades
Tempo
Importador/Exportador
Agente de Navegação
Operador Portuário
Nº Navios
Quantidade de Mercadoria

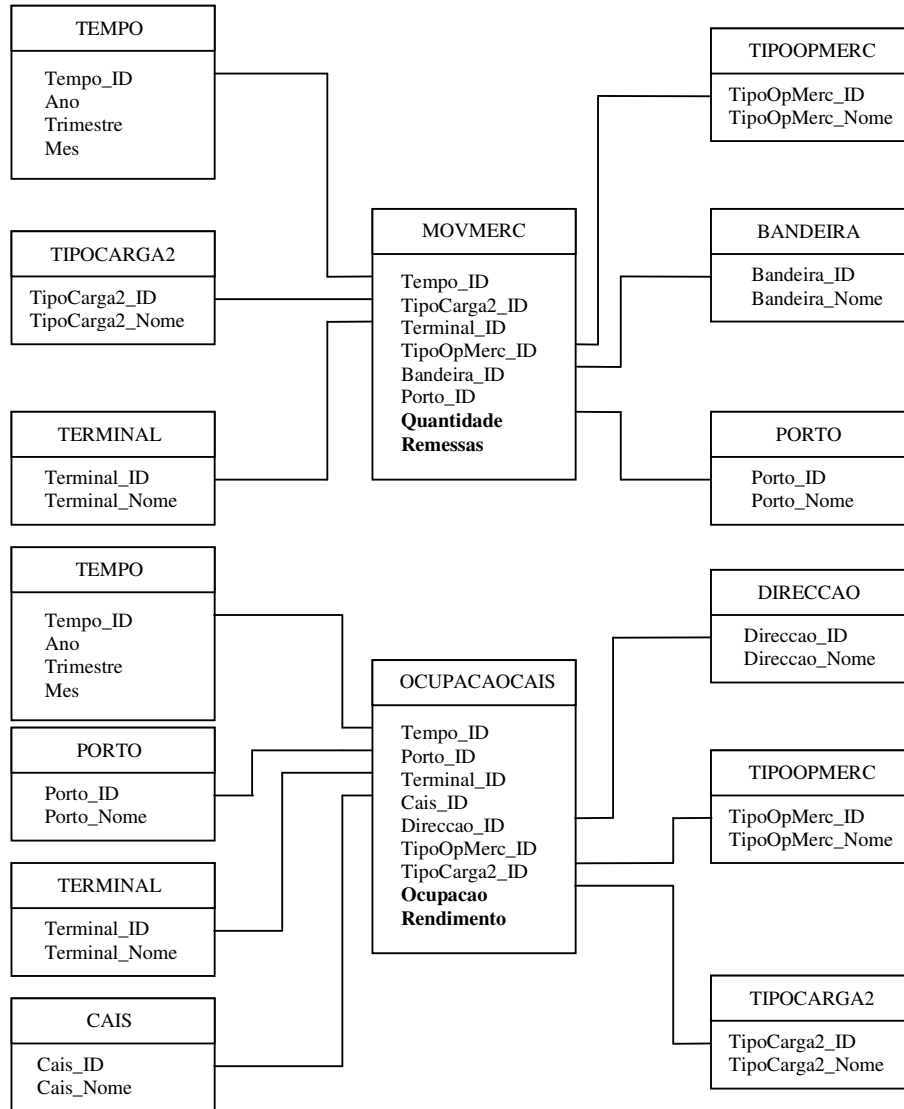
Taxa de Ocupação de Cais

Ocupação de Cais
Tempo
Porto
Terminal
Cais
Tipo de Operação de Navio
Tipo de Operação de Mercadoria
Tipo de Carga (2)
Ocupação
Rendimento

A.2 Modelo Multidimensional (base de dados dmporDW)







A.3 Glossário

Agregação	Forma de melhorar o desempenho das <i>queries</i> . Os factos são somados para as dimensões seleccionadas a partir da tabela de factos original. A tabela resultante dos valores agregados tem menos linhas resultando num melhor desempenho de <i>queries</i> .
Célula	Uma célula é definida pela intersecção de todas as dimensões de um cubo.
Cubo	É a unidade fundamental para armazenamento e visionamento da informação num sistema OLAP.
Cubo Virtual	É um cubo definido como parte de um outro cubo ou ainda resultado de dois ou mais cubos.
Data Mart	Um <i>data mart</i> tem a mesma definição de <i>data warehouse</i> mas consiste num pequeno conjunto de dados de uma organização utilizado para a área de <i>business analysis queries</i> num único departamento ou grupo de trabalho da organização.
Data Warehouse	Um <i>data warehouse</i> é um repositório de informação orientada por assuntos (<i>subject-oriented</i>), integrada, não volátil, variável no tempo e disponível para a realização de <i>queries</i> , análises e relatórios.
Data Warehousing	O processo de desenho, construção e manutenção de um sistema baseado numa <i>data warehouse</i> .
Dimensão	A mesma categoria de informação. É uma perspectiva que pode ser utilizada na consulta e análise dos dados de um cubo.
Drill Down	A operação de <i>Drill Down</i> traduz-se na operação inversa à operação de <i>Roll Up</i> , ou seja, permite descer um nível na hierarquia da dimensão em causa. Desta forma obtém-se uma visão mais granular da informação.
Drill Up	O mesmo que <i>Roll Up</i> .
Esquema em estrela	Forma comum na modelação multidimensional. Num esquema em estrela cada dimensão é representada por uma única tabela de dimensão.
ETL	<i>Extraction, Transformation, Loading</i> . Passagem da informação de uma localização para outra com possíveis transformações.
Hierarquia	Uma hierarquia define um caminho de navegação para as operações de <i>Roll Up</i> e de <i>Drill Down</i> , ou seja, diversas relações pai-filho entre níveis da mesma dimensão.

HOLAP	<i>Hybrid OLAP</i> . Os sistemas HOLAP armazenam os dados em bases de dados relacionais e as agregações em estruturas multidimensionais.
Metadados	Dados acerca de dados. Por exemplo, o número de tabelas de uma base de dados é um tipo de metadados.
Métrica	Valores mensuráveis armazenados em cada célula de um cubo.
Modelo multidimensional	Tipo de modelação de dados utilizado em <i>data warehousing</i> . Num modelo multidimensional existem dois tipos de tabelas: as tabelas de dimensão e as tabelas de factos.
MOLAP	<i>Multidimensional OLAP</i> . Os sistemas MOLAP armazenam os dados em cubos multidimensionais.
OLAP	<i>On-Line Analytical Processing</i> . Tecnologia que permite um método rápido de análise de dados históricos armazenados.
Package	É uma unidade de programação executável que corresponde ao objecto de mais alto nível no modelo de objectos do DTS. Um <i>package</i> contém quatro <i>collections</i> de objectos: ligações, variáveis globais, tarefas e <i>steps</i> .
Reporting	Tecnologia que permite a produção de relatórios sobre dados armazenados segundo o formato em papel ou mesmo segundo o formato HTML para visualização na Web.
ROLAP	<i>Relational OLAP</i> . Os sistemas ROLAP armazenam os dados em bases de dados relacionais.
Roll Up	A operação de <i>Roll Up</i> permite que o utilizador suba um nível de abstracção nos dados, ou seja, agrupa os dados respeitantes a determinada dimensão um nível acima na sua hierarquia. A esta operação também se designa por vezes <i>Drill Up</i> .
Tabela de dimensão	As tabelas de dimensão armazenam registos relacionados com a própria dimensão. Numa tabela de dimensão não são armazenados factos.
Tabela de factos	Tipo de tabela na modelação multidimensional. Uma tabela de factos inclui tipicamente dois tipos de colunas: colunas de factos (<i>fact columns</i>) e colunas com ligações directas para as dimensões.

A.4 Referências

- [Agrawal97] Agrawal, Rakesh; Gupta, Ashish; Sarawagi, Sunita. “Modeling Multidimensional Databases”, Proceedings of the Thirteenth International Conference on Data Engineering Pages: 232 – 243, 1997
- [Biere03] Biere, Mike. “Business Intelligence for the Enterprise”, Prentice Hall PTR, IBM Press, 2003
- [Bonczek81] Bonczek, R.H.; Holsapple, C.W.; Whinston, A.B. “Foundations of Decision Support Systems”, New York, Academic Press, 1981.
- [Boyno03] Boyno, Edward A. “Extraction, Transformation, and Loading in a Data Warehouse Course” Proc ISECON 2003 (San Diego)
- [Brandel01] Brandel, Mary. “Masters of Business Intelligence”, Computerworld 2/26/2001, Vol. 35 Issue 9, p61
- [Chaudhuri] Chaudhuri, Surajit. “An Overview of Data Warehousing and OLAP Technology”
- [Codd70] Codd, E. F. “A Relational Model of Data for Large Shared Data Banks”, Communications of the ACM, Vol. 13, No. 6, June 1970
- [Codd93] Codd, E.F.; S.B. Codd, C.T. Salley. “Providing OLAP (On-Line Analytical Processing) to User Analyst: An IT Mandate.”, <http://www.arborsoft.com/OLAP.html>
- [Contour-] Contour Components, “30 Ideas of Using OLAP”, http://www.contourcomponents.com/whitepapers/30_ideas_of_using_olap.htm
- [CrystalDecisions03] “Crystal Enterprise Report Application Server”, Overview, Crystal Decisions, 2003, http://support.businessobjects.com/communityCS/TechnicalPapers/cr9_ras.pdf
- [Cutshall05] Cutshall, R. “Introduction to Decision Support Systems”, Texas University, 2005
- [DataWarehousing-] Datawarehousing resource site, <http://www.1keydata.com/datawarehousing/etl.html>

-
- [David02]** David, L. Hamil. PMP, Mesa Solutions Inc., “An Introduction to Data Warehousing”, GITA 2002 UC Seminar Paper.
<http://www.mesasolutions.com/datawarehousing.html>
- [Dubler02]** Carl Dubler and Colin Wilcox, Microsoft Corporation, “Just What Are Cubes Anyway? (A Painless Introduction to OLAP Technology)”, April 2002,
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnexcl2k2/html/odc_DA_WhatRCubes.asp
- [Eblocks-]** Enterprise Blocks Home Page.
<http://www.eblocks.com/default.aspx>
- [EUROST94]** Directiva de normalização de estatísticas de transportes de mercadorias e de passageiros por mar, EUROSTAT – União Europeia. 1994.
- [Fazlollahi97]** Fazlollahi, Bijan; Parikh, Mihir A.; Verma, Sameer. “Adaptative decision support systems”, 1997, Decision Support Systems 20 (1997) 297-315
- [Gartner-]** Gartner Group Home Page.
<http://www.gartnergroup.com>
- [GestBarragens-]** gestBarragens - Sistema Integrado de Gestão da Informação para o Controlo de Segurança de Barragens, Grupo de Sistemas de Informação, INESC-ID,
<http://berlin.inesc-id.pt/alb/GestBarragens@82@-1.aspx>
- [Greyner01]** Greyner, Lynn. “Business Intelligence”, Computing Canada 3/16/2001, Vol. 27 Issue 6, p13, 2001
- [Harris98]** Harris, Robert. “Introduction to Decision Making”, VirtualSalt, 2 July, 1998,
<http://www.virtualsalt.com/crebook5.htm>
- [Hoffer02]** Hoffer, Jeffrey A.; Prescott, Mary B.; McFadden, Fred R. “Modern Database Management”, 6th Edition, Prentice Hall 2002
- [Howson04]** Howson, Cindi. “BI Scoreboard: OLAP”, Courtesy of Intelligent Enterprise Magazine, May 15th, 2004,
<http://www.bizintelligencepipeline.com/20900483>
- [Hu04]** Hu, Xiaohua; Cercone, Nick. “A Data Warehouse/Online Analytic Processing Framework for Web Usage Mining and Business Intelligence Reporting”

-
- [Inmon96]** Inmon, W. H. “Building the Data Warehouse”, 2nd Ed. John Wiley, New York, 1996
- [Inmon98]** Inmon, Bill. “Wherefore warehouse?”, Byte.com Jan1998, Vol. 23 Issue 1, Following page 88 p1
- [Inmon99]** W. H. Inmon. “Introduction to Data Marts”, 1999, <http://www.billinmon.com/library/articles/introdm.asp>
- [Jones98]** Jones, Katherine. “An Introduction to Data Warehousing: What Are the Implications for the Network?”, International Journal of Network Management, Vol. 8, 42–56, 1998
- [Kimball98]** R. Kimball, L. Reeves, M. Ross and W. Thornthwaite. “The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses”. Canada: John Wiley & Sons, 1998.
- [Larsen00]** Diane Larsen, Euan Garden. Microsoft Corporation, “Data Transformation Services (DTS) in Microsoft SQL Server 2000”. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsq12k/html/dts_overview.asp
- [Laudon02]** Laudon, Kenneth C.; Laudon, Jane P. “Management Information Systems – Managing the Digital Firm”, 7th Edition, Prentice Hall, 2002
- [Microsoft-]** Microsoft Corporation, ADO MD Object Model
- [MicroStrategy-]** MicroStrategy, Inc., “The 5 Styles of Business Intelligence: Industrial Strength Business Intelligence”, a White Paper prepared by MicroStrategy, Inc.
- [Morton78]** Morton, Scott; Keen, P.G.W. “Decision Support Systems: An Organizational Perspective”, Reading, MA: Addison-Wesley, 1978
- [Oliveira03]** Oliveira, Sérgio Vasconcelos. “Crystal Reports – Curso Completo”, FCA – EDITORA DE INFORMÁTICA LDA. – Março 2003, www.fca.pt
- [OMG-]** Datawarehousing, CWM and MOF Resource Page. <http://www.omg.org/cwm>
- [Peterson00]** Peterson, Tim; Pinkelman, Jim. “Microsoft OLAP Unleashed”, Sams Publishing, 2000

-
- [Planibase94]** SIGEP (Sistema de Gestão Portuária) 1.1, Guia do Utilizador, Planibase 1994
- [Power04]** Power, D.J. “Decision Support Systems Web Tour”. World Wide Web, version 4.3, January 11, 2004, <http://dssresources.com>
- [RDLSpec03]** Report Definition Language Specification, Third Draft, December 2003, Microsoft Corporation, <http://download.microsoft.com/download/4/7/d/47d7d117-9f91-49ad-98d5-46aa6f3251a8/RDLDec03.pdf>
- [ReportingServices-]** Microsoft SQL Server: Reporting Services Product Overview, <http://www.microsoft.com/sql/reporting/productinfo/overview.asp>
- [Santos05]** Santos, Paulo; Silva, Alberto R., “Produção Dinâmica e Interactiva de Relatórios Baseados em XML/RDL, com Foco na Geração de Queries MDX”, 3º XATA (XML - Aplicações e Tecnologias Associadas), Braga, Vila Verde, 2005
- [SAS-]** SAS | ETLQ, “ETLQ – Robust ETL with integrated data quality”, <http://www.sas.com/technologies/dw/etl/index.html>
- [Tan03]** Tan, Xin; Yen, David C.; Fang, Xiang. “Web warehousing: web technology meets data warehousing”, *Technology in Society* 25 (2003) 131–148
- [Thomsen99]** Thomsen, Erik; Spofford, George; Chase, Dick. “Microsoft OLAP Solutions”, John Wiley and Sons, 1999
- [Tolbert00]** Tolbert, D. CWM: “A Model-based Architecture For Data Warehouse Interchange”, Workshop on Evaluating Software Architectural Solutions (WESAS) 2000, UC Irvine, California, USA
- [Trujillo03]** Trujillo, Juan; Luján-Mora Sergio. “A UML based approach for modeling ETL processes in data warehouses”, *Conceptual Modeling - ER 2003, Proceedings Lecture Notes in Computer Science* 2813: 307-320, 2003
- [Vaihdiv04]** Vaihdiv, Rustam; Kersten, Gregory E. “Decision station: situating decision support systems”, *Decision Support Systems* 38 (2004) 283–303
- [Whitney03]** Whitney, Russ. *Data Discovery*, June 2003, “OLAP helps with the human parts of decision-making”

<http://www.winnetmag.com/SQLServer/Article/ArticleID/38719/38719.html>