

Produção Dinâmica e Interactiva de Relatórios Baseados na Linguagem XML/RDL, com Foco na Geração de Queries MDX

Paulo Santos¹, Alberto Rodrigues da Silva²

¹ INESC-ID e Instituto Superior Técnico,
Rua Alves Redol, n° 9 –1000-029 Lisboa, Portugal
paulo.m.santos@netcabo.pt

² INESC-ID e Instituto Superior Técnico,
Rua Alves Redol, n° 9 –1000-029 Lisboa, Portugal
alberto.silva@acm.org

Resumo. As aplicações de *Business Intelligence* têm por objectivo satisfazer as necessidades de informação das organizações e auxiliar nos seus processos de tomada de decisão. As capacidades de *reporting* enquadram-se na generalidade de sistemas de informação, mas com maior relevância nesta área do *Business Intelligence*. O aparecimento da nova plataforma de *reporting* da Microsoft, “*Reporting Services*”, veio disponibilizar uma plataforma extensível para construção e distribuição de relatórios especificados num formato aberto em XML. Neste artigo descreve-se um método de produção dinâmica e interactiva de relatórios baseados no modelo de objectos dos “*Reporting Services*”, definidos numa linguagem XML, mais precisamente na linguagem RDL (*Report Definition Language*).

1 Introdução

Desde os anos 90 as aplicações de *Business Intelligence* sofreram uma evolução enérgica graças ao crescimento exponencial da informação, aliado aos avanços tecnológicos. As aplicações de *Business Intelligence* permitem às organizações obterem o conhecimento necessário à melhoria dos seus processos de negócio, responderem atempadamente aos seus clientes e adaptarem-se a alterações no mercado [1]. Este conjunto de aplicações abrange todas as áreas funcionais de uma organização, com especial ênfase nos níveis estratégico e de gestão. Estas aplicações servem no auxílio à tomada de decisões, pelo que, se podem classificar em sistemas de suporte à decisão (*Decision Support Systems*), embora, também possam existir a nível estratégico [2][3]. Existem diversos tipos de aplicações *Business Intelligence*. Há quem defenda que são cinco os seus tipos de aplicação [4]. A área de *reporting* é uma dessas aplicações do *Business Intelligence*.

No âmbito das ferramentas e plataformas de *reporting* a Microsoft lançou recentemente a sua plataforma, “*SQL Server Reporting Services*”. Dentro desta área, existem diversos produtos no mercado, tais como, o MicroStrategy, o Business Objects,

ou o Cognos. Porém, esta é uma solução inovadora que permite a criação, gestão e distribuição de relatórios sob diferentes formatos (e.g., DOC, PDF, HTML, Excel) [5]. Ao passo que a maioria dos produtos existentes no mercado definem os relatórios segundo um formato proprietário, os Reporting Services definem-nos segundo um formato aberto. Uma característica fundamental desta plataforma consiste no facto dos seus relatórios serem definidos com base na linguagem RDL (*Report Definition Language*), que é um esquema XML. Este facto permite a disponibilidade de uma plataforma extensível para a criação e distribuição de relatórios especificados num formato aberto em XML.

Neste artigo descreve-se um método de produção dinâmica e interactiva de relatórios que tiram partido das capacidades da plataforma “*Reporting Services*”, e em particular da linguagem RDL.

Na Secção 2 introduz-se a arquitectura da plataforma “*Reporting Services*”. A Secção 3 introduz sucintamente a linguagem RDL (*Report Definition Language*) descrevendo o respectivo modelo de objectos. Nas Secções 4 e 5 é apresentado, respectivamente, a arquitectura do sistema proposto no âmbito da nossa investigação para a produção dinâmica e interactiva de relatórios definidos na linguagem RDL, e descrito o método para produção dinâmica e interactiva dos relatórios. Finalmente, na Secção 6 são apresentadas as conclusões.

2 Arquitectura da Plataforma “*Reporting Services*”

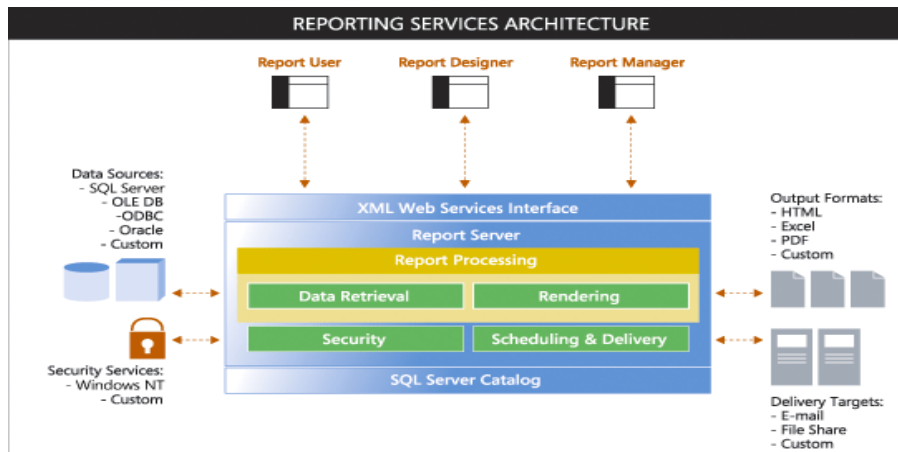


Figura 1 - Arquitectura dos Reporting Services (adaptado de [5])

Os *Reporting Services* possuem uma arquitectura modular, como sugerido na Figura 1. A plataforma tem por base o *Report Server* cuja função consiste em suportar toda e qualquer interacção dos utilizadores. Para além deste componente, existe ainda um conjunto de características que conferem à plataforma um carácter extensível. Esta arquitectura vai ser sumariamente apresentada nas secções seguintes.

2.1 Extensibilidade da Plataforma

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

O mesmo se passa relativamente aos formatos de *output*: para além dos formatos disponibilizados por omissão (i.e., 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.

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.

2.2 Componentes Aplicacionais

O *Report Server*, como componente principal da arquitectura, funciona como um serviço normal em 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*” (ver abaixo), 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 a 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.

3 A linguagem RDL (*Report Definition Language*)

A linguagem RDL (*Report Definition Language*) é a linguagem utilizada pelos “*Reporting Services*” para definição de relatórios. Todos os relatórios, segundo a plataforma de *reporting* da Microsoft, são descritos por um ficheiro em XML. Estes

ficheiros estão de acordo com um esquema XML (*XML schema*) e podem ser visualizados pelo servidor de *reporting* a partir do momento que estejam publicados no mesmo. A definição de um relatório à custa da linguagem XML, permite a criação de objectos específicos de acordo com o interesse do utilizador (ou programador) sem estar restringido a um conjunto limitado de objectos. Este aspecto prova a extensibilidade da plataforma, tanto a nível dos objectos do relatório (*design*) como dos tipos de fontes de dados que se possam aceder.

De acordo com o esquema XML para definição de relatórios actualmente publicada pela Microsoft [6] um relatório contém informações de acesso a dados e informações de *layout* do relatório.

Report XML Diagram

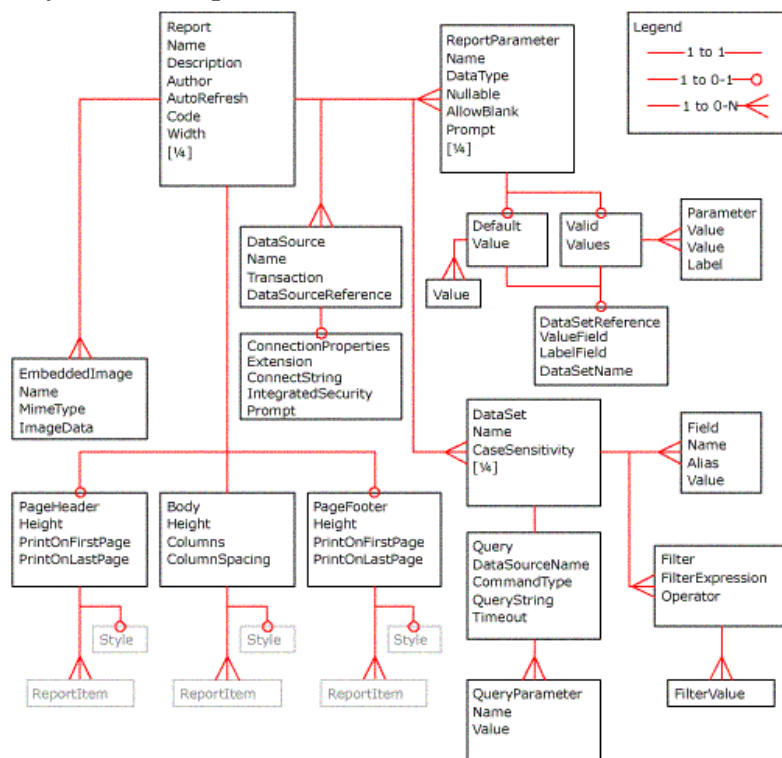


Figura 2 – Visão geral do modelo de objectos e de relatórios segundo a linguagem RDL

A Figura 2 apresenta o diagrama que descreve um relatório XML. Um relatório contém três elementos: *datasources*, *datasets* e *body*.

O elemento *datasources* descreve o tipo de dados e respectivas propriedades necessárias para acesso aos dados (e.g., nome da base de dados, nome de utilizador, palavra-passe). Embora só possa existir um elemento *datasources*, este pode conter vários elementos *datasource* que descrevem a informação referida anteriormente. Os “Reporting Services”, ao contrário de outras plataformas de *reporting*, permitem a definição de diversas fontes de dados para o mesmo relatório.

O elemento *datasets* contém uma lista de elementos *dataset*. Cada elemento *dataset* contém uma *query* de acesso a dados e um conjunto de campos disponíveis para o relatório. Cada elemento *dataset* contém ainda referência a um *datasource* definido anteriormente.

O elemento *body* contém a descrição do *layout* do relatório. Este elemento contém um sub-elemento (*ReportItems*) que contém por sua vez, os objectos a constar no relatório. Segundo o esquema XML [7], um relatório pode conter tabelas, matrizes, caixas de texto, imagens, linhas, rectângulos e ainda sub-relatórios. A Figura 3 apresenta o excerto de um ficheiro XML de definição de um relatório visualizado através do *Internet Explorer*.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Report xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://schemas.microsoft.com/sqlserver/reporting/2003/10/reportdefinition">
+ <DataSources>
- <DataSets>
  - <DataSet Name="DataSet1">
    + <Fields>
    - <Query>
      <DataSourceName>dmpport</DataSourceName>
      <CommandText>SELECT {[Measures].[Nnavios]} on columns, NON EMPTY Descendants([Tempo].[All Tempo],
        [Tempo].[Mes], LEAVES) on rows, NON EMPTY {[ClasseDWT].[Classedwt Nome].Members} on pages FROM
        [EuroF1]</CommandText>
    </Query>
  </DataSet>
</DataSets>
- <Body>
  - <ReportItems>
    + <Textbox Name="Textbox_Title">
    + <Matrix Name="Matrix1">
    + <Chart Name="Chart1">
  </ReportItems>
  <Height>26.25cm</Height>
  <ColumnSpacing>1cm</ColumnSpacing>
  <Style />
</Body>
  <Width>21cm</Width>
</Report>
```

Figura 3 - Exemplo de um ficheiro XML (RDL) para definição de um relatório

No exemplo da Figura 3 é possível verificar a definição dos elementos *datasources* e *datasets*. O elemento *datasets* contém apenas um *dataset*. O elemento *ReportItems* contém três objectos: uma caixa de texto, uma matriz e um gráfico.

4 Arquitectura do sistema proposto

Esta investigação tem como requisito básico a produção de relatórios *on the fly*, isto é, de forma dinâmica. Em vez dos relatórios serem desenhados previamente e definidos de modo estático, pretendemos que seja o próprio utilizador a definir o seu relatório, interactivamente (em *runtime*), consoante as suas necessidades de informação.

A produção de relatórios nesta solução foca-se essencialmente na geração de *queries* MDX [8][9], ou seja, *queries* de acesso a dados multidimensionais. A utilização

de dados multidimensionais neste protótipo traduz-se num caso particular, sendo por isso, a solução extensível a outros tipos de dados. Embora a solução apenas se foque na produção dinâmica de *queries* MDX, também se poderia ter dado maior importância na geração dinâmica do *design* do relatório – o sistema desenvolvido apenas permite a construção dinâmica do título do relatório e escolher entre a visualização do relatório sob a forma de tabela ou a forma gráfica. Contudo, este aspecto, não menos importante na produção de relatórios, cai fora do âmbito da nossa investigação.

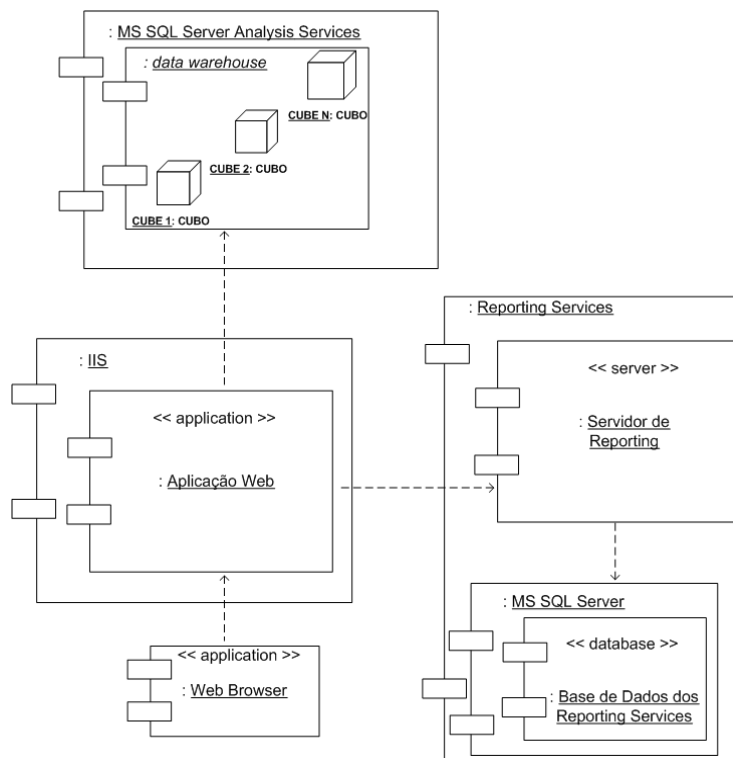


Figura 4 - Diagrama de componentes da arquitectura do sistema proposto

Este sistema é composto por três elementos fundamentais: (1) base de dados multi-dimensional; (2) aplicação Web; e (3) plataforma de *reporting*.

A Figura 4 ilustra a arquitectura do sistema proposto.

A **base de dados multidimensional** (*data warehouse*) contém todos os cubos com a informação necessária à produção de relatórios e está armazenada no servidor *Analysis Services* [10].

A **aplicação Web** reside num servidor aplicacional IIS (*Internet Information Services*).

A **plataforma de reporting** utilizada foi os “*Reporting Services*” (descritos na Secção 2). Esta plataforma contempla um servidor de relatórios (*Report Server*) e uma base de dados.

A aplicação pode ser acedida por um utilizador através de um *browser*, por exemplo, o *Internet Explorer*.

Embora actualmente todos os elementos do sistema residam na mesma máquina, tanto a aplicação *Web*, a base de dados multidimensional, como a plataforma de *reporting* podem residir em servidores distintos. Se tal situação acontecer, basta alterar o ficheiro de configurações da aplicação (*Web.config*) de modo a contemplar os endereços dos respectivos servidores.

5 Método para produção dinâmica e interactiva de relatórios

Embora os “*Reporting Services*” disponibilizem um componente específico para realizar o *design* de relatórios (o *Report Designer* como foi já referido na sub-secção 2.2), a produção dinâmica de relatórios obriga-nos a recorrer a API’s específicas dos *Reporting Services* e à utilização de uma linguagem de programação (utilizou-se nesta investigação o *C#* e o ambiente *Visual Studio .Net 2003*).

A produção dinâmica e interactiva de relatórios subdivide-se em dois problemas: (1) a selecção da informação a constar no relatório (carácter interactivo); e (2) a construção do relatório propriamente dito (carácter dinâmico).

Lista de Cubos Disponíveis: Quadro EuroF1 - Movimento de Navios

Dimensões do Cubo: Tipo de Navio

Esta dimensão só dispõe de uma hierarquia

Níveis da Hierarquia: Tipo de Navio Eixo para o nível: Colunas OK

Membros do Nível: Abastecimento off shore, Actividades off shore, Batelão de carga seca aberta, Batelão de carga seca coberta Eixo para os membros: Slicer OK

Elementos das Colunas: [Tempo],[Ano] Ano Esconder Elementos Vazios:

Elementos das Linhas: [ClasseDWT],[ClasseDWT Nome] Classe de Deadweight Esconder Elementos Vazios:

Elementos Slicer: [TipoNavio],[Batelão de carga sec] Batelão de carga seca aberta

Elementos Parâmetro: [Direcção],[Direcção Nome] Direcção

Elementos Medidas: [Measures],[Nnavios],[Measures] Nnavios, Totaldwt

Figura 5 - Protótipo desenvolvido

5.1 Interactividade

Relativamente ao carácter interactivo, a solução adoptada passou pela construção de um protótipo que disponibilizasse a informação existente na base de dados, de modo a que o utilizador pudesse seleccionar os dados pretendidos para o relatório. Embora neste sistema tenham sido utilizados dados multidimensionais, extraídos a partir dos *Analysis Services*, a solução estende-se com maior facilidade também a bases de dados relacionais, e por isso, é generalizável. A Figura 5 apresenta um *snapshot* do protótipo desenvolvido, em que um utilizador constrói interactivamente o relatório.

5.2 Dinâmica

Em relação ao carácter dinâmico dos relatórios, a solução consiste no mapeamento das entidades seleccionadas pelo utilizador no modelo de objectos dos “*Reporting Services*” (descritos na Secção 3).

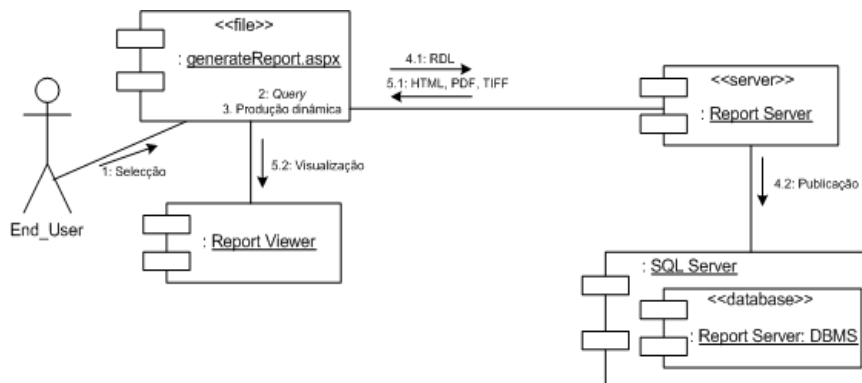


Figura 6 - Diagrama de componentes relativo à produção de um relatório dinâmico

A produção dinâmica e interactiva de relatórios desenrola-se segundo a seguinte sequência de passos (conforme sugerido na Figura 6):

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

Neste passo o utilizador tem a possibilidade de escolher o cubo que deve constar no relatório e quais as respectivas dimensões do cubo. 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, por forma a facilitar a construção da *query* MDX.

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

Este passo consiste na geração da *query* MDX de acordo com as dimensões indicadas pelo utilizador. Como exemplo do MDX gerado apresentamos o seguinte excerto (relacionado com os dados do caso de estudo), fazendo referência à cláusula *Columns* que apenas contém Medidas (*Measures*).

```
SELECT NON EMPTY {[Measures].[Pesobruto]} on columns,  
NON EMPTY Descendants([Tempo].[All Tempo], [Tempo].[Mes], LEAVES) on  
rows,  
NON EMPTY CROSSJOIN ({[Porto].[Porto  
Nome].Members},{[TipoCarga1].[TipoCarga1 Nome].Members}) on pages  
FROM [EuroA1]
```

Produção do relatório (Passo 3)

A produção de um relatório segundo os “*Reporting Services*” consiste na criação de um ficheiro XML com a extensão RDL (*Report Definition Language*). Uma vez que se pretende a produção dinâmica de relatórios o ficheiro XML teve que ser criado de raiz, programaticamente. A produção do relatório implica a criação das fontes de dados, das *queries* MDX e dos objectos de visualização (no caso, tabelas e/ou gráficos), de acordo com as opções do utilizador.

Na produção do relatório segundo a linguagem RDL, recorreu-se à ferramenta *xsd.exe* (*XML Schema Definition Tool*) que acompanha a *framework .Net*. Esta ferramenta permite, a partir de um esquema XSD (*XML Schema Definition Language*) a geração de classes *C#* conformes ao esquema. Foi utilizado o esquema de dados *XSD* que acompanha os “*Reporting Services*” e que descreve a especificação RDL [6].

Uma vez tendo as classes geradas, é possível a criação dos objectos que constarão no relatório, de acordo com as opções do utilizador. A geração das classe permite programar num nível mais elevado, ao invés de programar directamente sobre XML. Após a criação de todos os objectos é efectuada a sua serialização para disco, ou seja, é criado o ficheiro de extensão RDL (a Figura 7 apresenta um excerto de código XML produzido).

```

- <Matrix Name="Matrix1">
  <Top>2cm</Top>
  <Left>0.5cm</Left>
  <DataSetName>DataSet1</DataSetName>
+ <Corner>
- <ColumnGroupings>
  - <ColumnGrouping>
    <Height>0.63cm</Height>
    - <DynamicColumns>
      - <Grouping Name="Matrix_Grouping_Tempo_Ano">
        - <GroupExpressions>
          <GroupExpression>=Fields!Tempo_Ano.Value</GroupExpression>
        </GroupExpressions>
      </Grouping>
    + <ReportItems>
      <Visibility />
    </DynamicColumns>
  </ColumnGrouping>
+ <ColumnGrouping>
+ <ColumnGrouping>
+ <ColumnGrouping>
</ColumnGroupings>
+ <RowGroupings>
+ <MatrixRows>
+ <MatrixColumns>
</Matrix>

```

Figura 7 - Excerto do código XML produzido para definição de uma matriz

Publicação do relatório (Passo 4)

Uma vez tendo sido produzido o relatório no passo 3 é necessário publicá-lo para o servidor para que este possa ser visualizado. 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*). 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);

```

A função auxiliar *getXMLValue* permite obter o valor de um parâmetro definido no ficheiro de configurações *Web.config*.

Visualização do relatório publicado através do *Report Viewer* (Passo 5)

Finalmente, o passo 5 envolve a visualização do relatório num ambiente *Web*. Para tal, foi necessário utilizar o *viewer* desenvolvido pela Microsoft para o efeito, o **Report Viewer**.

```
protected Microsoft.Samples.ReportingServices.ReportViewer ReportViewer1;  
...  
ReportViewer1.ReportPath = "/" + ut.getXMLValue("reportsPath") + "/" +  
    ut.getXMLValue("reportName");  
ReportViewer1.ServerUrl = "http://" + ut.getXMLValue("datasource") +  
    ut.getXMLValue("reportServer");
```

A visualização do relatório no âmbito da nossa investigação é possível segundo duas formas: a forma em tabela (matriz) e a forma gráfica. A Figura 8 apresenta a visualização de um relatório através do Report Viewer sob a forma em tabela.

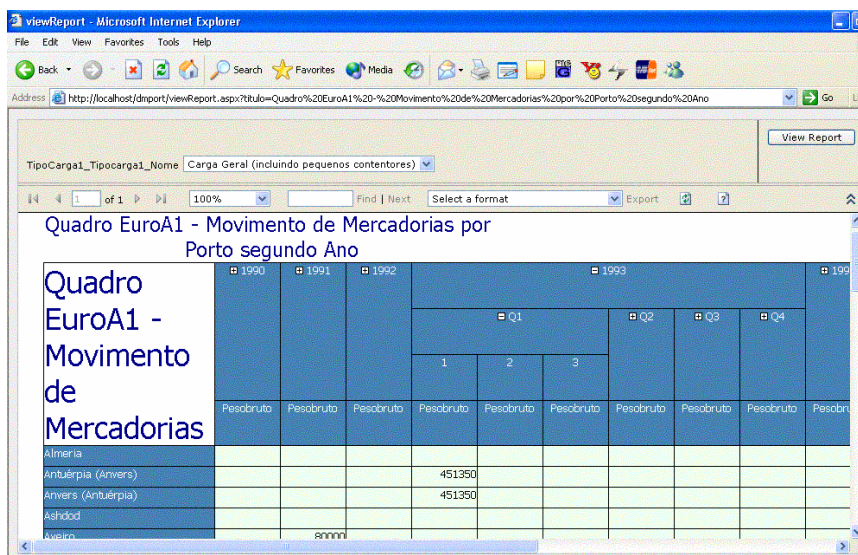


Figura 8 - Visualização do relatório através do *Report Viewer*

6 Conclusões

A plataforma de *reporting* da Microsoft, “*Reporting Services*”, permite disfrutar de uma plataforma extensível para construção e distribuição de relatórios. Ao contrário da maioria dos produtos existentes no mercado na área de *reporting*, 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 de acordo com as necessidades do seu negócio. A esta vantagem acresce o facto de podermos implementar dinamicamente relatórios (em *runtime*), sem estarmos restringidos a um conjunto dos mesmos definidos previamente (em *design-time*).

Referências

1. Acuma, World Class Information Management, <http://www.acuma.co.uk/802568AC005A7933/pages/BusinessIntelligenceOverviewBusinessIntelligence.html>
2. Laudon, Kenneth C.; Laudon, Jane P., “Management Information Systems – Managing the Digital Firm”, 7th Edition, Prentice Hall, 2002
3. Jessup, Leonard; Valacich, Joseph, “Information Systems Today”, “Chapter 6 – Organizational Information Systems”, Prentice Hall, 2003
4. “The 5 Styles of Business Intelligence: Industrial-Strength Business Intelligence”, A White Paper prepared by MicroStrategy, Inc., Copyright 2002, <http://www.microstrategy.com/Solutions/5StylesBook.asp>
5. Microsoft SQL Server: Reporting Services Product Overview, <http://www.microsoft.com/sql/reporting/productinfo/overview.asp>
6. Microsoft SQL Server 2005, Report Definition Language Schema, May 2004, <http://schemas.microsoft.com/sqlserver/reporting/2003/10/reportdefinition/reportdefinition.xsd>
7. 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>
8. Spofford George, “MDX Solutions with Microsoft SQL Server Analysis Services”, Wiley, 2001
9. Whitehorn, Mark; Pasumansky, Mosha; Zare, Robert, “Fast Track to MDX”, Springer
10. Jacobson, Reed, “Microsoft SQL Server 2000 Analysis Services Step by Step”, Wiley, 2001, Microsoft Press, 2000

(Nota: As hiperligações foram acedidas no período compreendido entre Outubro e Novembro de 2004)