

Uma Ferramenta WEB para apoio à Decisão em Ambiente Hospitalar

Mikael de Souza Fernandes ¹, Gustavo Zanini Kantorski¹²

mikael@cpd.ufsm.br, gustavoz@cpd.ufsm.br

¹ Curso de Sistemas de Informação, Universidade Luterana do Brasil (ULBRA), Campus Santa Maria, RS, Brasil.

² Centro de Processamento de Dados, Universidade Federal de Santa Maria (UFSM), Santa Maria, RS, Brasil.

Resumo: OLAP e data warehouse são tecnologias essenciais para auxiliar a tomada de decisão. Estas surgiram para superar as limitações de sistemas OLTP. A maioria das soluções OLAP e data warehouse são proprietárias que dificultam ou inviabilizam o processo de integração. Este artigo apresenta uma ferramenta web, *open source*, para auxílio aos gestores da área de saúde.

Palavras-chave: *data warehouse, OLAP, Web, saúde.*

1. Introdução

A utilização de ferramentas de suporte à decisão no apoio aos gestores de universidades, principalmente de Hospitais Universitários, permite uma agilidade na obtenção de informações estratégicas sobre o negócio para que possam se adequar às mudanças e atender as necessidades da instituição e da comunidade de modo satisfatório. Os sistemas para informatização dos procedimentos transacionais de Hospitais Universitários normalmente não possuem ferramentas de análise.

Neste enfoque foi desenvolvido um sistema de informações hospitalares para o Hospital Universitário de Santa Maria. Este sistema, disponibilizado na *web*, permite aos gestores do hospital analisarem as informações sobre a abrangência de atuação do hospital na região central do estado do Rio Grande do Sul (Brasil), além da comprovação da efetividade dos procedimentos de baixa, média e alta complexidade realizados pelo hospital.

O sistema foi desenvolvido através de tecnologias OLAP (*On-Line Analytical Processing*), cujo objetivo é transformar dados em informações capazes de dar suporte a decisões gerenciais de forma flexível e em tempo hábil. Desta forma,

OLAP precisa oferecer informações existentes, oportunas, precisas e inteligíveis [Thomsen 2002].

O presente artigo tem por objetivo apresentar uma ferramenta OLAP *Web*, para análise de informações da área da saúde, aplicadas ao Hospital Universitário de Santa Maria. Na próxima seção são apresentadas as tecnologias utilizadas para o desenvolvimento da ferramenta e na seção 3 é apresentada a arquitetura e a ferramenta desenvolvida. Trabalhos futuros e considerações finais são descritos na seção 4.

2. Tecnologia OLAP

OLAP é uma categoria de software específica para realizar processamento analítico dos dados de *data warehouse*, de maneira que este processamento deve ocorrer com alto desempenho, consistência e interatividade e auxiliar a tomada de decisão em uma organização através da interpretação desses dados em várias visões multidimensionais [Cood, Codd e Salley 1993].

As ferramentas OLAP são geralmente de uma das três arquiteturas: ROLAP – *Relational OLAP*, MOLAP – *Multidimensional OLAP* e HOLAP – *Hybrid OLAP*. ROLAP realiza o processamento do *data warehouse* em uma estrutura física do modelo relacional e modelada dimensionalmente. A arquitetura MOLAP utiliza o armazenamento físico multidimensional. E a arquitetura HOLAP integra as características de ROLAP e MOLAP, onde os dados atômicos são armazenados em um modelo relacional e os agregados e dimensões são guardados em uma matriz n-dimensional.

Independente da arquitetura utilizada, as ferramentas OLAP manipulam os dados do *data warehouse* através de um estrutura multidimensional denominada cubo. Os cubos representam o *data warehouse* e são organizados de maneira a realizar consultas rápidas e dinâmicas voltadas ao apoio à decisão. Os cubos são definidos através de dimensões e medidas. As dimensões fornecem as informações descritivas, enquanto que as medidas disponibilizam informações quantitativas que se deseja consultar [Kimball 1996].

A manipulação das estruturas multidimensionais pode ser realizada através de operações OLAP. As operações OLAP são o *drill-up* ou *roll-up*, na qual os dados são agrupados em um nível com maior granularidade, o *drill-down*, no qual existe o detalhamento dos dados para um nível com menor granularidade, o *slice & dice* que seleciona uma parte do cubo, o *Pivoting* que realiza a inversão /rotação dos eixos do cubo para visualização de uma consulta, o *Rank* que ordena as consultas de acordo com os valores das medidas e o *Drill-across* que realiza o detalhamento dos dados por meio de múltiplos níveis de dimensões diferentes.

O Mondrian e o JPivot são exemplos de ferramentas OLAP *open source* e que possuem interface para a web. O Mondrian é um servidor OLAP desenvolvido na linguagem Java sendo parte de um conjunto de tecnologias *open sources* pertencentes ao projeto *Pentaho Business Intelligence* [Pentaho 2007].

Para um melhor entendimento do funcionamento do servidor Mondrian é possível dividi-lo em 4 camadas [Mondrian 2007]: a camada de apresentação (*presentation layer*), a camada dimensional (*dimensional layer*), camada estrela (*star layer*) e a camada de armazenamento (*storage layer*). A tecnologia JPivot é responsável pela camada de apresentação, projetada para trabalhar com vários servidores OLAP, especialmente com o Mondrian. O JPivot é uma ferramenta que através de suas *tag librarys* permite apresentação e interação com os dados analisados [JPivot 2007], isto é, visualização e manipulação de tabelas, resultado de consultas OLAP. A segunda camada, dimensional, é a camada que recebe, analisa e envia consultas MDX (*Multidimensional Expressions*). A camada estrela é responsável pela ligação da consulta MDX recebida pela camada anterior e a execução da mesma sobre a base de dados mapeada. E por última, a camada de armazenamento que executa as consultas solicitadas pela camada estrela sobre a base de dados relacional, logo caracterizando o Mondrian como um servidor ROLAP.

3. A Ferramenta

Nesta seção é apresentada a ferramenta *Neuro Business Intelligence* que foi desenvolvida para atender as necessidades do hospital universitário. Na seção 3.1 são apresentados a modelagem dos cubos e o mapeamento realizado entre o modelo relacional e o modelo dimensional. Na seção 3.2 são mostradas a ferramenta desenvolvida e as tecnologias utilizadas.

3.1. Modelagem Multidimensional

O modelo de armazenamento multidimensional implica em dois tipos de tabelas: tabelas de dimensões e tabelas de fatos. Uma tabela de dimensões consiste em tuplas de atributos da dimensão. Uma tabela de fatos pode ser imaginada como se possuísse tuplas, uma para cada fato registrado. Esse fato contém alguma variável ou variáveis observadas ou mensuradas e identifica as mesmas como ponteiros para tabelas de dimensões. As tabelas de fatos contêm os dados e as dimensões identificam cada tupla nesses dados [Navathe 2000]. O *data warehouse* foi criado em um modelo ROLAP em uma base *MySQL Server*.

Para modelagem dos cubos foi utilizada uma ferramenta multidimensional denominada *Cube Designer* (Figura 1), que faz parte do projeto do servidor Mondrian, *Pentaho Business Intelligence* [Pentaho 2007]. A ferramenta de modelagem multidimensional realiza uma conexão via JDBC (*Java Database Connectivity*) com diversas bases de dados pré-configuradas, entre elas *MySQL*.

A ferramenta apresenta uma lista de tabelas do banco, onde é possível seleccionar e manipular as tabelas e seus atributos. Após define-se, dentre as tabelas selecionadas, a tabelas de fatos. Entretanto, na definição de dimensões, propriedades e medidas a ferramenta apresenta limitações nos recursos de configurações, onde não é possível utilizar-se toda potencialidade da modelagem. O resultado da modelagem é armazenado em um arquivo em XML (*eXtensible Markup Language*) que é manipulado pelo servidor Mondrian.

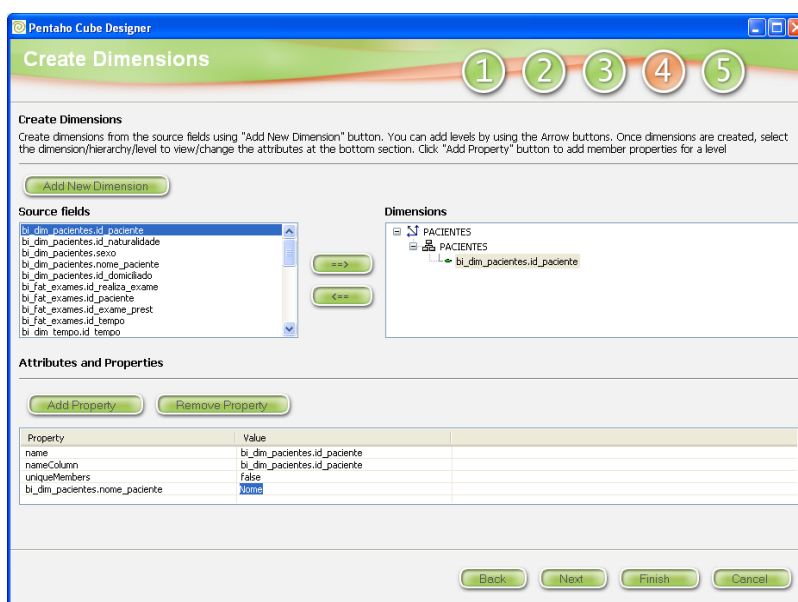


Figura 1. Ferramenta Cube Designer – modelagem de dimensões.

De acordo com [Brito 2004], a lógica do Mondrian é implementada através de *Schemas*, que definem o modelo multidimensional lógico e o mapeamento deste modelo em um modelo físico e relacional. O primeiro modelo, lógico, é formado de elementos definidos pelo Schema, sendo estes: cubo (*cube*), dimensão (*dimension*), hierarquias (*hierarchy*), níveis (*level*) e membros (*members*). Por conseguinte, o modelo físico é a base de dados relacional que é mapeada pelo modelo lógico através do *Schema*. Logo, para uma mais completa definição do Mondrian Schema é necessária uma manipulação no arquivo XML por meio de um editor de texto qualquer.

Na modelagem da ferramenta para a instituição de saúde focou-se na análise de dois aspectos: um deles referencia os exames realizados pelos pacientes, enquanto

o outro corresponde ao consumo de materiais farmacêuticos. A modelagem dimensional é apresentada na Figura 2.

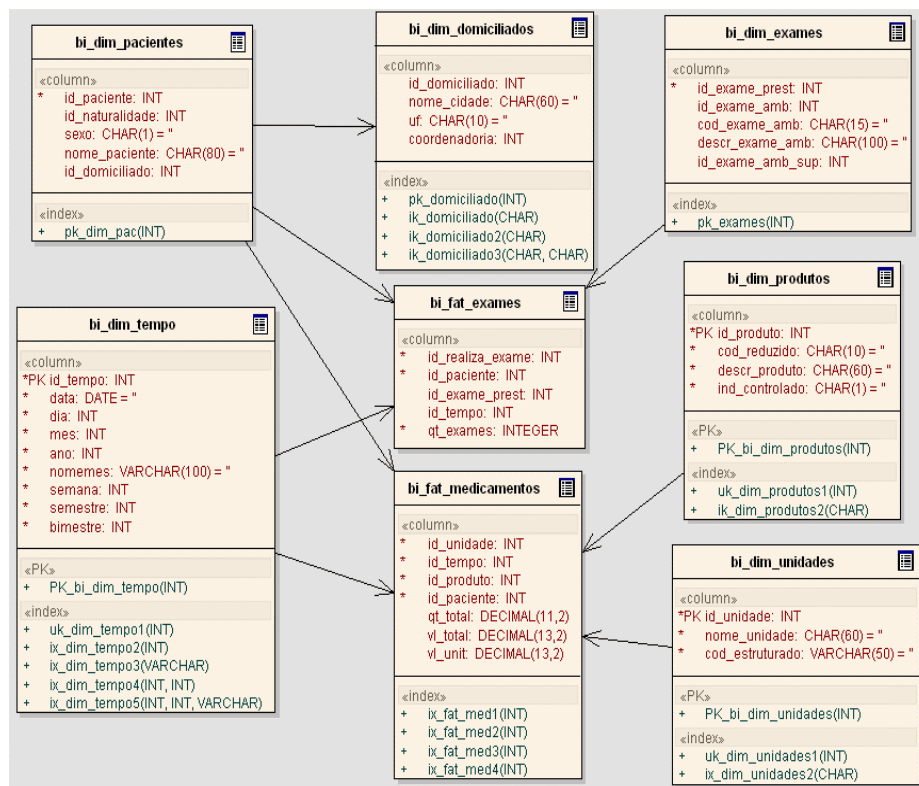


Figura 2. Modelagem Dimensional – Dimensões e Fatos.

A primeira modelagem refere-se aos procedimentos de exames cuja tabela de fatos é a *bi_fat_examenes* (Figura 2), com as respectivas medidas de número de pacientes e de número de exames realizados, possuindo como aspectos para análise as seguintes dimensões:

- Pacientes: dimensão que caracteriza basicamente duas propriedades dos pacientes do hospital, sexo e domicílio (identificando a cidade e estado do paciente). A propriedade domicílio é o resultado da *tag* do *Schema* denominada *joinTable*, referenciando as tabelas pacientes e domiciliados (Figura 3);
- Exames: dimensão que representa o código e a descrição dos exames;

- Tempo: dimensão que permite uma robusta análise dos fatos através das propriedades dia, mês, ano, semana, bimestre e semestre.

A Figura 3 mostra um fragmento do arquivo XML onde pode ser visualizado o mapeamento da dimensão Pacientes com a tabela de `bi_dim_pacientes` modelada na Figura 2 e o `join` realizado com a tabela `bi_dim_domiciliados` que representa a modelagem *snow-flake* da dimensão pacientes. É possível verificar também os quatro níveis que a dimensão pacientes possui, estado, cidade, sexo e nome. Além disso, pode-se verificar a tabela de fatos, `bi_fat_examenes`, que faz parte do cubo exames.

```
<Cube name="Exames">
  <Table name="bi_fat_examenes"/>
  <Dimension name="PACIENTES" foreignKey="ID_PACIENTE">
    <Hierarchy name="PACIENTE" hasAll="true" primaryKey="ID_PACIENTE" primaryKeyTable="bi_dim_pacientes">
      <Join leftkey="ID_DOMICILIADO" rightkey="ID_DOMICILIADO">
        <Table name="bi_dim_pacientes"/>
        <Table name="bi_dim_domiciliados"/>
      </Join>
      <Level name="Estado" table="bi_dim_domiciliados" column="UF" uniqueMembers="false"/>
      <Level name="Cidade" table="bi_dim_domiciliados" column="NOME_CIDADE" uniqueMembers="false"/>
      <Level name="Sexo" table="bi_dim_pacientes" column="SEXO" uniqueMembers="false"/>
      <Level name="Nome" table="bi_dim_pacientes" column="NOME_PACIENTE" uniqueMembers="false"/>
    </Hierarchy>
  </Dimension>
</Cube>
```

Figura 3. Mapeamento de Esquemas XML e a Tabela de Dimensões.

A outra modelagem corresponde à movimentação de materiais farmacêuticos pelos setores do hospital, denominados de unidades. A modelagem representa os valores financeiros e as devidas quantidades através de agregações e somas da tabela de fatos `bi_fat_medicamentos` (Figura 2). Para análise de tais medidas, além das dimensões citadas anteriormente (pacientes e tempo), novas dimensões foram modeladas:

- Unidades: dimensão que representa os setores que consomem os materiais da instituição, por meio das propriedades nome e código;
- Produtos: dimensão que caracteriza o código e a descrição dos diversos materiais referenciados na tabela de fatos.

É importante salientar que, embora existam vários cubos modelados, as dimensões são compartilhadas entre os diversos cubos, evitando re-trabalho na modelagem. Assim, por exemplo, a dimensão tempo é a mesma que é utilizada em todos os cubos modelados.

3.2. A Ferramenta Neuro

Por meio da utilização das tecnologias Mondrian e JPivot, e ainda com o *Web container* Apache Tomcat, foi possível o desenvolvimento de uma nova ferramenta denominada Neuro. A ferramenta Neuro compreende um aplicativo Java baseado no servidor OLAP Mondrian com características novas agregadas para o Hospital Universitário de Santa Maria (HUSM). A Figura 4 apresenta a arquitetura definida para a ferramenta.

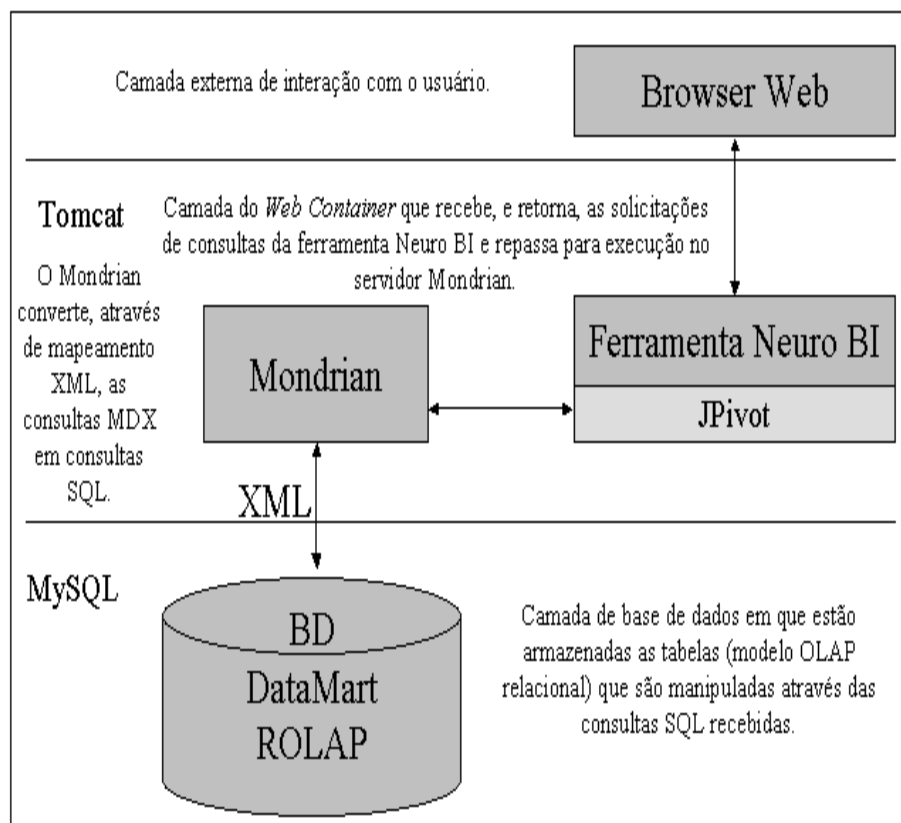


Figura 4. Arquitetura da Ferramenta.

O servidor OLAP Mondrian é a camada principal da aplicação Neuro BI, pois processa e retorna as consultas (MDX) geradas pela devida aplicação. Entretanto, para que haja esse processamento é necessária uma prévia configuração na

estrutura do servidor OLAP, primordialmente, após a geração do *Schema*, deve-se adicioná-lo ao diretório da aplicação, podendo conter ainda outros *Schemas*, no *container Web* e configurá-lo (no arquivo *datassources.xml*).

Na Figura 5 é mostrada a interface principal da ferramenta. No *menu*, à esquerda, podem ser visualizados os esquemas modelados. Na parte central são mostradas informações sobre o período dos dados gerados e os cubos disponíveis para consulta. Na Figura 5 é possível visualizar o cubo de exames e o cubo de produtos farmacêuticos. Novos cubos e consultas podem ser modelados e adicionados à interface e disponibilizados para consulta. Na versão atual da ferramenta o período de referência para geração dos cubos é o mesmo, não sendo possível à geração de períodos diferentes para cubos diferentes.

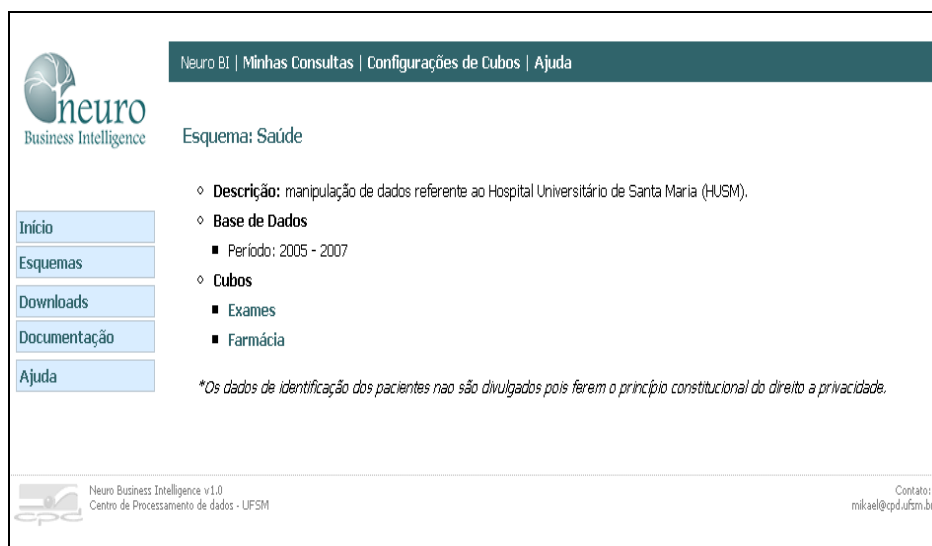


Figura 5. Interface Inicial da Ferramenta.

Depois de configurado o Mondrian *Schema*, é possível gerar as consultas que estarão contidas em arquivos no formato JSP (*Java Server Pages*) em um diretório específico (diretório *queries*), sendo que cada arquivo possui uma única consulta. Um arquivo possui, além da convencional *tag* Java (*core*) na qual é especificada o título para a consulta, a *tag* denominada *mondrianQuery* referente ao *JPivot*. Nesta *tag* é necessário especificar os parâmetros JDBC para conexão, o *Schema* a ser manipulado e, em seu corpo, a consulta MDX para acesso aos dados, conforme a Figura 6. Logo, existe uma multiplicidade de arquivos do formato JSP onde cada um identifica uma única consulta MDX, sendo que o nome de cada arquivo corresponderá ao parâmetro na execução de cada consulta.

As consultas mdx mostradas na Figura 6 compreendem uma parte da geração e exibição da consulta. Outras funcionalidades (através de botões), como cabeçalho e rodapé estarão presentes em um outro arquivo do formato JSP presente no diretório *root* (raiz) da aplicação. Desta forma a manutenção da devida aplicação refletirá em todas as consultas que fazem uso.

```
<%@ page session="true" contentType="text/html; charset=ISO-8859-1" %>
<%@ taglib uri="http://www.tonbeller.com/jpivot" prefix="jp" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>

<jp:mondrianQuery id="query01" jdbcDriver="com.mysql.jdbc.Driver"
    jdbcUrl="jdbc:mysql://amargo:3306/osbi"
    catalogUri="/WEB-INF/queries/husm.mondrian.xml">

select {[Measures].[vl_total], [Measures].[qt_total]} ON COLUMNS,
NON EMPTY Hierarchize(Crossjoin(Crossjoin([TEMPO].[All Tempo].children,
    {[PACIENTES.PACIENTE]}), {[PRODUTOS.PRODUTO].[All PRODUTOS]})) ON ROWS
from [farmacia]

</jp:mondrianQuery>

<c:set var="title01" scope="session">Período/Cidades/Produtos </c:set>
```

Figura 6. Arquivo JSP com parâmetro de conexão e exemplo de consulta mdx.

A Figura 7 mostra o retorno, em um Browser Web, da consulta MDX mostrada na Figura 6. Na interface é possível visualizar ao topo, alguns *links* da ferramenta Neuro. Logo abaixo, o título da consulta, especificado no arquivo JSP da consulta (através da *tag* denominada *core*). Acima da tabela de dados gerada, os botões que correspondem às operações OLAP permitidas pelo servidor Mondrian. E no centro, a tabela com o resultado da consulta MDX gerada, listando inicialmente as dimensões e medidas como colunas e os respectivos membros como linhas. É mostrada uma operação *drill member* sobre a hierarquia (*hierarchy*) pacientes da dimensão (*dimension*) de mesmo nome. E finalmente, abaixo da tabela, é possível retornar ao devido esquema (*Schema*) em que a consulta pertence, por meio de um link, e o rodapé da ferramenta.

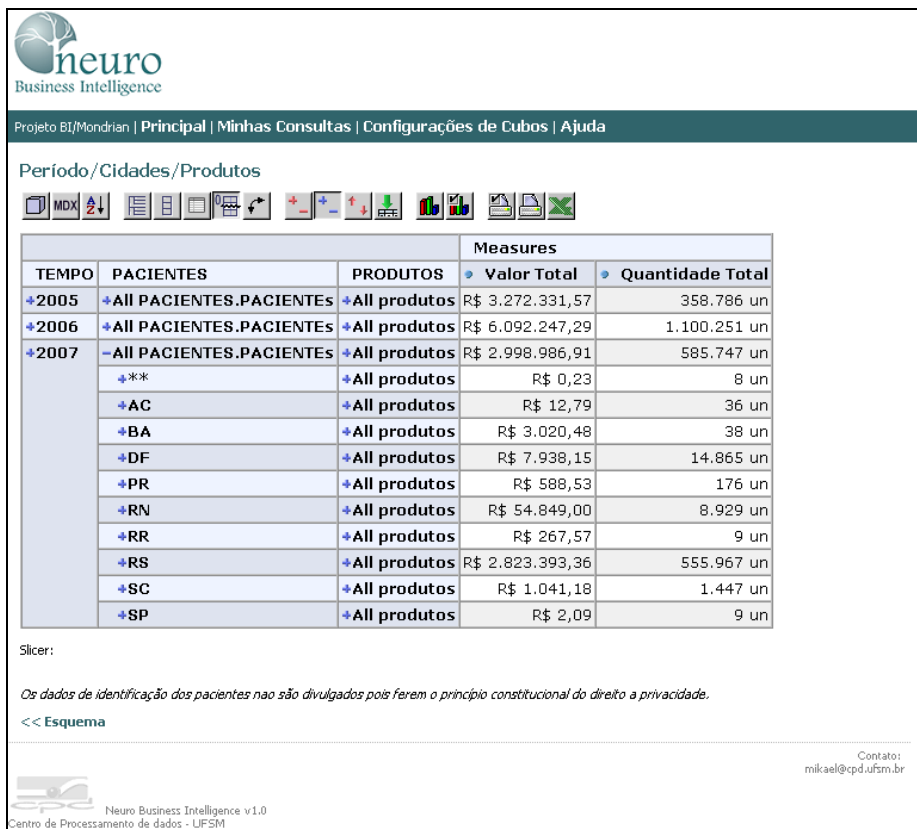


Figura 7. Interface de Retorno de uma Consulta.

4. Conclusões

Este artigo apresenta uma ferramenta OLAP *web, open source*, para apoio aos gestores de hospitais no auxílio à tomada de decisões. Um estudo de caso foi realizado no hospital universitário de Santa Maria, RS, onde foram detectadas várias situações relativas à realização de exames pelos pacientes e o consumo de materiais farmacêuticos de pacientes internados.

As ferramentas de consultas OLAP limitam-se, embora nem todas, a soluções privadas que prejudicam o processo de integração. A ferramenta desenvolvida neste trabalho utiliza tecnologias *open source* e realiza a integração dessas

tecnologias. A ferramenta utiliza uma interface mais amigável com a qual os usuários possam trabalhar e realiza a integração entre tecnologias existentes como o Mondrian e o JPivot.

Como trabalhos futuros destacam-se a necessidade dos usuários poderem configurar e salvar as consultas para que não seja necessária, a cada execução de consulta, a configuração e formatação das informações geradas. Além dessas configurações deve-se trabalhar melhor algumas modificações para que o próprio gestor utilize a ferramenta ao invés de delegar essa geração de consultas para usuários especializados.

Referências

- Brito, Maiquel de. (2004). Proposta de um Data Warehouse de informações acadêmicas. Instituto de Ciências Exatas e Tecnológicas, Centro Universitário Feevale, Novo Hamburgo.
- Cood, E.F., Cood S.B., Salley, C.T. (1993). Providing OLAP (OnLine Analytical Processing) to user-analysts: IT Mandate, White Paper, Arbor Software Corporation.
- Kimball, R. (1996). The Data Warehouse Toolkit. John Wiley.
- Navathe, S. e Elmasri, R. (2000). Fundamentals of Database Systems, Addison Wesley, 3th edition.
- JPivot. (2007.) A JSP based OLAP, <http://jpivot.sourceforge.net>, December.
- Mondrian. (2007). Mondrian OLAP Server, <http://mondrian.sourceforge.net>, December.
- Pentaho. (2007). Open Source Business Intelligence, <http://www.pentaho.org>, December.
- Thomsen, E. (2002). OLAP Solutions: Building Multidimensional Information Systems, 2nd ed., John Wiley & Sons, New York, NY, .