



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
ESCOLA DE INFORMÁTICA APLICADA

UMA COMPARAÇÃO ENTRE PLATAFORMAS DE BUSINESS INTELLIGENCE  
USANDO BASES DE DADOS GOVERNAMENTAIS

Filipe Trindade de Lima

Jean Rouberte de Freitas

**Orientador**

Asterio Kiyoshi Tanaka

RIO DE JANEIRO, RJ – BRASIL

JANEIRO DE 2014

UMA COMPARAÇÃO ENTRE PLATAFORMAS DE BUSINESS INTELLIGENCE  
USANDO BASES DE DADOS GOVERNAMENTAIS

Filipe Trindade de Lima

Jean Rouberte de Freitas

Projeto de Graduação apresentado à Escola de  
Informática Aplicada da Universidade Federal do  
Estado do Rio de Janeiro (UNIRIO) para obtenção do  
título de Bacharel em Sistemas de Informação.

Aprovada por:

---

Prof. Asterio Tanaka, D.Sc. (UNIRIO)

---

Prof. Claudia Cappelli, D.Sc. (UNIRIO)

---

Prof. Fernanda Baião, D.Sc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

JANEIRO DE 2014

## **Agradecimentos**

### Agradecimentos Filipe

Agradeço a todos que me suportaram ao longo dos anos que estive na faculdade.

Agradeço aos meus pais, familiares e amigos pelo apoio mesmo nas horas mais difíceis.

Agradeço ao meu amigo e braço direito Jean Freitas pela paciência, parceria e incentivo, que foram determinantes para a conclusão desta monografia.

Agradeço ao professor e orientador Asterio Tanaka, que foi peça fundamental para a realização deste trabalho com sua sabedoria, prestatividade e disponibilidade.

Agradeço também a Deus, onde encontrei as forças necessárias para levar essa jornada com tranquilidade.

## Agradecimentos Jean

Agradeço primeiramente a Deus, que me permitiu completar esta etapa da minha vida.

Aos meus pais, meu irmão e minha família que sempre me apoiaram durante toda a minha vida, acreditaram no meu potencial e me incentivaram nos momentos difíceis, tornando possível a conclusão desta etapa.

Aos meus amigos da UNIRIO que comigo dividiram experiências, enfrentaram dificuldades e compartilharam momentos de alegria, facilitando a conclusão desta jornada.

Aos professores e funcionários da Escola de Informática Aplicada e, em especial, ao orientador Asterio Kiyoshi Tanaka pelo aprendizado, incentivo, paciência e apoio que possibilitaram a conclusão deste trabalho.

Ao meu companheiro nesse trabalho Filipe, que teve a competência, determinação e paciência necessários para a conclusão deste trabalho.

## RESUMO

As organizações estão buscando cada vez mais investir em soluções de Business Intelligence (BI), em especial as empresas privadas, para obter vantagem competitiva em relação aos seus concorrentes. BI ajuda a analisar informações consolidadas para permitir tomadas de decisão mais acertadas, a partir de dados coletados de diversos sistemas de informação disponíveis na organização ou fora dela. Com base neste fato, surgiu a ideia de usar ferramentas de BI, aplicando sobre um estudo de caso as técnicas de modelagem e desenvolvimento de uma aplicação analítica. O trabalho serviu como prova de conceito para a comparação de plataformas de BI com representatividade no mercado, utilizando métricas preestabelecidas pelo Gartner Group<sup>1</sup>, no que diz respeito à integração, entrega de informação e análise. Optou-se pela escolha de duas plataformas de universos distintos: uma plataforma *open source* (Pentaho BI Suite Community Edition) e uma outra plataforma com licença proprietária (Oracle BI Enterprise Edition – OBIEE). Além da comparação das ferramentas segundo as métricas do Gartner Group, o desenvolvimento da aplicação analítica permitiu comparar, também, as características do processo de desenvolvimento em cada uma das plataformas.

**Palavras-chave:** Business Intelligence, Dados abertos governamentais, Pentaho, OBIEE.

---

1 Gartner Group 2013: Magic Quadrant for Business Intelligence and Analytics Platforms

## **ABSTRACT**

Organizations are increasingly looking to invest in Business Intelligence (BI) solutions, especially private companies to obtain a competitive advantage over its competitors. BI helps analyze consolidated information to allow wiser decision making, based on data collected from various information systems available in the organization or outside it. Based on this fact, the idea of using BI in a case study techniques for modeling and development of an analytical application. The work served as a proof of concept for the comparison of BI platforms represented on the market, using pre-established metrics by the Gartner Group, with respect to integration, information delivery and analysis. We opted for the choice of two platforms in distinct worlds: an open source (Pentaho BI Suite Community Edition) platform and another platform with proprietary license (Oracle BI Enterprise Edition – OBIEE). Besides comparing the tools according to Gartner metrics, the development of analytical application allowed also to compare the characteristics of the development process on each platform.

**Keywords:** Business Intelligence, Open Government Data, Pentaho, OBIEE.

## Índice

1	Introdução .....	1
1.1	Motivação.....	2
1.2	Objetivos .....	3
1.3	Organização do texto.....	4
2	Conceitos de Business Intelligence.....	6
2.1	Data Warehouse .....	6
2.2	Modelagem Normalizada .....	7
2.3	Modelagem Desnormalizada ou Dimensional .....	8
2.4	Definição das Métricas do Gartner Group para Avaliação de Plataformas de BI	12
3	Descrição e Comparação das Plataformas OBIEE e Pentaho.....	16
3.1	Oracle Business Intelligence Enterprise Edition 11g.....	16
3.2	Pentaho BI Suite.....	25
3.3	Aplicação das Métricas no Pentaho e no OBIEE.....	34
3.4	Tabelas Comparativas .....	40
4	Estudo de Caso: A Base de dados da Copa 2014.....	42
4.1	Introdução .....	42
4.2	Processo de ETL .....	43
4.3	Modelagem Dimensional – Esquema Estrela .....	44
5	Desenvolvimento no OBIEE.....	49
5.1	Criando o repositório.....	49
5.2	Análises, <i>Dashboards</i> e Relatórios .....	56
5.3	Alertas e Agentes .....	64
5.4	Segurança .....	65
6	Desenvolvimento no Pentaho .....	70
6.1	Conexão com o banco de dados .....	70
6.2	Criação do repositório .....	71

6.3 Criando metamodelo e análises.....	78
6.4 Alertas e ações.....	83
6.5 Segurança .....	84
7 Comparação do Desenvolvimento no OBIEE e no Pentaho.....	87
7.1 Criação de repositório .....	87
7.2 Medidas Calculadas .....	88
7.3 Criação de Hierarquias.....	89
7.4 Dados Pré-Agregados.....	89
7.5 Utilização de Variáveis/Parâmetros .....	89
7.6 Medidas de Séries Temporais .....	90
7.7 Análises, Relatórios e Dashboards.....	90
7.8 Alertas e Ações .....	91
7.9 Segurança .....	92
7.10 Outras Observações.....	92
8 Conclusões .....	93
8.1 Considerações Finais.....	93
8.2 Trabalhos Futuros .....	94
Bibliografias Consultadas.....	95
Anexo I DDL para as Tabelas de Dimensão e Fato .....	99

## **Índice de Tabelas**

Tabela 1: <i>Checklist</i> da categoria Integração .....	40
Tabela 2: <i>Checklist</i> da categoria Entrega de Informação .....	41
Tabela 3: <i>Checklist</i> da categoria Análise.....	41
Tabela 4: Descrição da Dimensão Empreendimento.....	45
Tabela 5: Descrição da Dimensão Instituição .....	46
Tabela 6: Descrição da Dimensão Tempo .....	47
Tabela 7: Descrição da Dimensão Andamento Financeiro .....	47
Tabela 8: Descrição da tabela Fato Execução Financeira .....	48

## Índice de Figuras

Figura 1: Ilustração de uma tabela fato .....	9
Figura 2: Ilustração de uma tabela de dimensão .....	9
Figura 3: Modelagem de um esquema floco de neve .....	11
Figura 4: Modelagem de um esquema estrela .....	12
Figura 5: Diagrama da arquitetura do OBIEE .....	17
Figura 6: Diagrama da arquitetura do OBIEE - BI Clients .....	17
Figura 7: Diagrama da arquitetura do OBIEE - Presentation Services e Presentation Catalog.....	19
Figura 8: Diagrama da arquitetura do OBIEE - Oracle BI Scheduler e Scheduler Database.....	20
Figura 9: Diagrama da arquitetura do OBIEE - BI Server, Admin Tool e BI Repository .....	21
Figura 10: Interface do Administration Tool.....	22
Figura 11: Camadas de um repositório e fluxo de dados gerado por uma consulta .....	23
Figura 12: Diagrama da arquitetura do OBIEE - Datasources .....	24
Figura 13: Relatório no Pentaho Community Edition.....	26
Figura 14: Relatório no Pentaho Enterprise Edition.....	26
Figura 15: Representação da arquitetura do Pentaho .....	28
Figura 16: Representação de uma transformação no Kettle.....	31
Figura 17: Arquitetura do Pentaho Analysis Services.....	32
Figura 18 - Interface do Aggregation Designer.....	34
Figura 19: Esquema estrela usado no projeto.....	45
Figura 20: Campos para conexão utilizando OCI .....	49
Figura 21: Tabelas Físicas e Tabelas <i>Alias</i> na camada física.....	50
Figura 22: Esquema estrela inicial do repositório .....	50
Figura 23: Camada de Modelo de Negócios e Mapeamento.....	51
Figura 24: Camada de Apresentação .....	52
Figura 25: Dimensão Lógica de Tempo .....	53
Figura 26: Camada de Modelo de Negócios e Mapeamento - Esquema Estrela.....	54
Figura 27: Exemplo de medida de séries temporais .....	56
Figura 28: Portal do OBIEE .....	57
Figura 29: Aba <i>Criteria</i> da área de criação de análises.....	58
Figura 30: Aba <i>Results</i> da área de criação de análises .....	59

Figura 31: Exemplo de Análise Criada no OBIEE.....	60
Figura 32: Área de Edição de <i>Dashboards</i> .....	61
Figura 33: Exemplo de <i>Dashboard</i> criado no OBIEE.....	62
Figura 34: Relatório criado no modo guiado.....	63
Figura 35: Relatório editado no editor de relatórios.....	64
Figura 36: Ícone de alerta destacado .....	65
Figura 37: Janela contendo a lista de Alertas .....	65
Figura 38: Análise filtrada através de segurança.....	69
Figura 39 - Definindo parâmetros para conexão no Mondrian .....	70
Figura 40 - Definindo parâmetros para conexão no Pentaho BI Server .....	71
Figura 41 - Adicionando uma tabela Fato ao cubo.....	72
Figura 42 - Adicionando dimensões ao cubo .....	73
Figura 43 - Adicionando as métricas ao cubo .....	74
Figura 44 - Adicionando campo calculado.....	74
Figura 45 - Criação de medida calculada .....	75
Figura 46 - Desenvolvimento de tabelas agregadas no Aggregation Designer.....	76
Figura 47 - Publicando o repositório .....	77
Figura 48 - Criação do metamodelo de negócios .....	79
Figura 49 - Relatório gerado pelo Pentaho Report Designer.....	80
Figura 50 - Análise simples no Saiku Analytics.....	81
Figura 51 - Análise gráfica no Saiku Analytics.....	82
Figura 52 – Exemplo de criação de dashboard no Pentaho Community.....	83
Figura 53 - Criação de Action Sequences no Pentaho Design Studio.....	84
Figura 54 - Definições de papéis e usuários no Pentaho BI Server.....	85
Figura 55 - Alteração de permissão de segurança nos metadados .....	86

# 1 Introdução

Com o avanço tecnológico iniciado a partir da segunda metade do século XX, a capacidade de armazenamento e a velocidade de processamento de dados aumentaram significativamente. Diversas organizações possuíam grande quantidade de dados, porém não existia uma metodologia adequada que permitisse a análise eficiente destes dados.

Com o aumento da competitividade entre as organizações, a necessidade de obtenção de informações a partir da enorme massa de dados se tornou cada vez maior. As organizações que possuíam informações mais precisas e pertinentes iriam ter maiores chances de se fortalecer e aumentar sua participação no mercado.

Os sistemas utilizados até o início da década de 1990 não permitiam que análises complexas fossem realizadas. Esses sistemas, chamados de OLTP (Online Transaction Processing), têm como objetivo armazenar os dados provenientes da operação da empresa. Os usuários de sistemas transacionais são responsáveis pela execução operacional do dia-a-dia da empresa, como registrar pedidos, cadastrar novos clientes e registrar reclamações. Esses usuários quase sempre lidam com um registro de dados por vez. Executam as mesmas tarefas operacionais diversas vezes, de modo repetitivo.<sup>2</sup> Sistemas OLTP possuem características específicas para que o armazenamento de dados corrente seja realizado de maneira eficiente e consistente.

Algumas características de sistemas OLTP:

- Armazenamento de valores correntes e detalhados
- Organização de dados por aplicação
- Frequentes atualizações de dados
- Estrutura de dados relacional

---

<sup>2</sup> Kimball, Ralph; Ross, Margy. The Data Warehouse Toolkit. John Wiley, 2002.

- Uso estruturado e repetível. Não permite análise e descoberta de informação <sup>3</sup>

Para suprir a necessidade de análise de informações de maneira eficiente, foram desenvolvidos novos conceitos como Data Warehouse e modelagem dimensional. Esses conceitos são utilizados por sistemas OLAP (On-line Analytical Processing) que possibilitam que análises de um grande volume de dados sejam feitas. Os usuários que utilizam sistemas OLAP veem os dados de maneira consolidada, raramente lidam com um registro por vez. As consultas exigem que centenas ou milhares de linhas sejam pesquisadas e agregadas para gerar um conjunto de respostas. Para aumentar ainda mais a complexidade, as consultas dos usuários mudam frequentemente para que as respostas desejadas sejam obtidas<sup>4</sup>.

Algumas características de sistemas OLAP:

- Armazenamento de valores históricos e sumarizados
- Organização dos dados por assunto, negócio. Abrangência ampla
- Dados raramente são atualizados
- Estrutura de dados dimensional
- Consultas realizadas sob diferentes perspectivas permitindo descoberta de informação
- Desempenho otimizado para análises complexas<sup>5</sup>

## 1.1 Motivação

A análise de informações confiáveis e consistentes em conjunto com um planejamento estratégico bem elaborado é uma prática chave para o sucesso de uma organização. O fornecimento destas informações, bem como sua obtenção e organização é responsabilidade de *Business Intelligence* (traduzindo literalmente, Inteligência de Negócios). Esta área da Tecnologia da Informação utiliza sistemas OLAP para permitir

---

<sup>3</sup> <http://www.uniriotec.br/~tanaka/SAIN/01-VisaoGeral.pdf>

<sup>4</sup> Kimball, Ralph; Ross, Margy. *The Data Warehouse Toolkit*. John Wiley, 2002.

<sup>5</sup> <http://www.uniriotec.br/~tanaka/SAIN/01-VisaoGeral.pdf>

uma maior precisão no processo de tomada de decisão e no desenvolvimento organizacional como um todo.

Os benefícios reais do uso desses sistemas e melhores tomadas de decisão são perceptíveis em aspectos do negócio como: melhor rentabilidade, redução de custos, maior eficiência, crescimento na participação do mercado, maior satisfação de funcionários e clientes ou quaisquer outros objetivos que uma organização possa ter.

Atualmente, soluções de BI são utilizadas por diversas instituições e organizações, desde empresas de telefonia ou mercado varejista até instituições de ensino como colégios e universidades, e órgãos de governo. Cada tipo de organização requer um planejamento estratégico baseado em indicadores próprios, que devem ser monitorados e analisados para apoiar tomadas de decisão.

Um exemplo prático, à parte do que será discutido neste projeto, é a utilização de sistemas de BI em instituições bancárias. Essas organizações possuem diversos indicadores específicos tais como: quantidade de clientes inadimplentes, índice de evasão de clientes, tempo médio de atendimento em call-centers.

Tendo em vista os benefícios perceptíveis do uso de BI nas organizações em geral, surge a motivação de comparar duas plataformas de BI conceituadas no mercado. Foram feitos dois tipos de comparação. A primeira comparação utiliza métricas definidas pelo Gartner Group, permitindo identificar pontos fortes e fracos de cada plataforma. A segunda comparação tem como objetivo identificar as características do processo de desenvolvimento de uma aplicação de BI. Para isso, foram aplicadas técnicas de modelagem e desenvolvimento em um estudo de caso. Para essa comparação foi escolhida uma base de dados governamental devido ao fácil acesso a esta através da internet.

## **1.2 Objetivos**

O objetivo do trabalho é comparar duas plataformas de BI de duas formas. A primeira comparação foi feita utilizando métricas definidas pelo Gartner Group. A segunda comparação foi realizada através do desenvolvimento de uma aplicação de BI utilizando dados reais provenientes da base de dados abertos governamentais disponibilizada pelo Governo Federal através do site "Portal da Transparência – Copa

2014" (<http://www.portaltransparencia.gov.br/copa2014/>). O objetivo é comparar o processo de desenvolvimento de uma aplicação de BI nas duas plataformas.

Para o desenvolvimento da aplicação de BI, foi realizada uma modelagem multidimensional a partir do estudo da base de dados disponível no mencionado sítio, para a criação de um Data Mart. Em seguida, foi desenvolvida uma aplicação de BI, como prova de conceito, utilizando duas ferramentas de BI diferentes, Oracle Business Intelligence Enterprise Edition e Pentaho BI Suite Community Edition.

### **1.3 Organização do texto**

O presente trabalho está estruturado em capítulos e, além desta introdução, está organizado da seguinte forma:

- Capítulo 2: Neste capítulo é explicada a importância do BI para uma organização, são apresentados conceitos de Business Intelligence como Data Warehouse e modelagem dimensional e são definidas as métricas do Gartner Group;
- Capítulo 3: Apresenta a descrição dos componentes e arquitetura das plataformas OBIEE e Pentaho e compara as plataformas de acordo com as métricas preestabelecidas;
- Capítulo 4: Neste capítulo é apresentada a base de dados da Copa 2014 utilizada no projeto, é explicado o processo de extração, transformação e importação dos dados e é descrito o esquema estrela do Data Mart criado para o desenvolvimento da aplicação;
- Capítulo 5: Neste capítulo é detalhado o desenvolvimento da aplicação BI utilizando o Oracle Business Intelligence Enterprise Edition;
- Capítulo 6: Neste capítulo é detalhado o desenvolvimento da aplicação BI utilizando o Pentaho BI Suite Community Edition;
- Capítulo 7: Neste capítulo são comparados os desenvolvimentos do OBIEE e do Pentaho, expondo as dificuldades e facilidades de cada solução. Também são levantadas algumas vantagens de uma ferramenta com relação à outra.

- Capítulo 8: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

## 2 Conceitos de Business Intelligence

O papel de Business Intelligence é coletar, organizar e disponibilizar dados provenientes de diferentes fontes para os seus respectivos usuários. Esta centralização das informações é feita através de inúmeros passos, desde a extração de dados dos sistemas através de um processo denominado ETL (Extract, Transform, Load), passando pela construção de um modelo para armazenar estes dados, até a presença de um *hardware* capaz de processar diversas consultas às bases analíticas.

### 2.1 Data Warehouse

O conceito de DW (Data Warehouse) é devido a Bill Inmon, que o definiu como “uma coleção de dados orientada a assunto, integrada, variante no tempo e não volátil, para apoio ao processo de tomada de decisões da gerência”. O DW é o banco de dados histórico da BI, responsável pelo armazenamento de toda informação relativa às atividades de uma organização. O ponto mais importante de um DW é a possibilidade de análise de grandes volumes de dados oriundos dos sistemas transacionais, integrando-os e atendendo à necessidade de múltiplos usuários finais. Com o amadurecimento de sistemas deste tipo a partir da década de 1990 e a impossibilidade dos sistemas OLTP gerarem simples relatórios, a implementação do DW tornou-se realidade nas grandes corporações.

São objetivos de um Data Warehouse:

- Fornecer fácil acesso às informações
- Apresentar informações consistentes
- Ser adaptável e resistente a mudanças
- Fornecer um ambiente seguro para proteção da informação

- Servir como fundação para a tomada de decisão<sup>6</sup>

Existem dois modelos principais para um Data Warehouse: modelo normalizado e modelo desnormalizado (dimensional).

## 2.2 Modelagem Normalizada

A modelagem normalizada proposta por Bill Inmon em seu livro *Building The Data Warehouse* consiste na criação de um Data Warehouse normalizado em Terceira Forma Normal (3FN), como um banco de dados relacional convencional.

A modelagem de dados ocorre em três níveis: modelagem de alto nível (chamada de *Entity Relationship Diagram*, ou ERD), modelagem intermediária (chamada de *Data Item Set*, ou DIS), e modelagem de baixo nível (chamada de *Physical Model*).

Na modelagem de alto nível, são criadas as entidades e seus relacionamentos. Representa o nível mais abstrato do modelo. As entidades que farão parte do modelo devem ser escolhidas previamente junto ao arquiteto, gerente e usuário final.

Na modelagem de nível intermediário, é criado um modelo intermediário para cada entidade identificada no modelo de alto nível. Esses modelos intermediários representam as entidades em uma visão mais detalhada.

A modelagem de baixo nível é feita a partir do modelo de nível intermediário. São criadas tabelas, chaves e características físicas do modelo. Após o término da construção do modelo, é determinada qual será a granularidade e o particionamento dos dados a fim de determinar otimizações de performance.<sup>7</sup>

As principais vantagens da modelagem normalizada são:

- Minimizar consumo de espaço em disco
- Diminuir redundâncias de dados
- Aumentar a consistência dos dados
- Diminuir o esforço com procedimentos de ETL

---

<sup>6</sup> Kimball, Ralph; Ross, Margy. *The Data Warehouse Toolkit*. John Wiley, 2002.

<sup>7</sup> Inmon, William. *Building the Data Warehouse*. Wiley Publishing, 2005.

As principais desvantagens são:

- Dificuldade para gerar consultas complexas
- Dificuldade de percepção das origens dos dados

## 2.3 Modelagem Desnormalizada ou Dimensional

Na modelagem desnormalizada ou dimensional, introduzida por Ralph Kimball, dados de transação são particionados em "fatos", que são normalmente dados numéricos. Além disso, são criadas "dimensões", que são os dados que dão contexto aos fatos. Por exemplo, uma transação de vendas pode ser dividida em fatos como quantidade de produtos vendidos e o preço de cada produto. Para dar contexto a esses fatos, podem ser criadas as dimensões de data, cliente, produto e localidade<sup>8</sup>.

### 2.3.1 Fatos

Fatos são tabelas que armazenam os indicadores de uma empresa. Uma tabela fato possui uma lista de indicadores e chaves estrangeiras para as dimensões. São exemplos de indicadores: preço, quantidade, receita bruta, despesas, custos, etc. Uma medição é feita a partir da interseção das dimensões com o indicador. Por exemplo, receita bruta do produto X vendido em janeiro de 2013 no Rio de Janeiro ao cliente Y.

Uma linha em uma tabela fato corresponde a uma medição e todas as medições estão em alguma tabela fato. As medições na tabela fato devem possuir a mesma granularidade<sup>9</sup>, isto é, devem ter o mesmo nível de detalhamento segundo as dimensões relacionadas.

As tabelas fato ocupam a maior parte do armazenamento de dados, em média, 90% do DW. Isto porque, sendo o DW um banco de dados histórico, ele armazena, em cada linha, uma ocorrência do fato que representa, podendo chegar a centenas de milhões de linhas, como no exemplo acima, todas as vendas de todos os produtos para todos os clientes em todas as cidades, a cada dia ao longo de um período de dezenas de anos.

---

<sup>8</sup> [http://en.wikipedia.org/wiki/Data\\_warehouse](http://en.wikipedia.org/wiki/Data_warehouse)

<sup>9</sup> Kimball, Ralph; Ross, Margy. The Data Warehouse Toolkit. John Wiley, 2002.

Fato_Vendas
FK_Produto
FK_Cliente
FK_Data
FK_Cidade
Quantidade
Preço

Figura 1: Ilustração de uma tabela fato

### 2.3.2 Dimensões

As tabelas de dimensão contêm a descrição textual do negócio. Cada coluna da tabela é um atributo da dimensão. Cada uma dessas tabelas pode conter muitos atributos, chegando a ter várias dezenas de colunas. Cada tabela de dimensão é definida por sua chave primária, que serve como base para a integridade referencial em operações de junção com as tabelas fato<sup>10</sup> relacionadas.

Produto
PK_Produto
Descrição
Modelo
Fabricante
Cor
Tamanho
Peso
Formato
Código de Barras

Figura 2: Ilustração de uma tabela de dimensão

<sup>10</sup> Kimball, Ralph; Ross, Margy. The Data Warehouse Toolkit. John Wiley, 2002.

As tabelas de dimensões não possuem muitas linhas, se comparadas com as tabelas de fatos, variando de algumas poucas (no exemplo acima, algumas dezenas de cidades ou algumas centenas de produtos) a alguns milhares (por exemplo, 3.650 dias em dez anos ou centenas de milhares de clientes), poucas vezes ultrapassando a casa dos milhões.

### **2.3.3 Esquema Floco de Neve**

Em um esquema floco de neve as tabelas de dimensões são normalizadas. Nele, diferentes tabelas fato se conectam compartilhando uma ou mais dimensões. Este tipo de estrutura é chamado de “floco de neve” porque as tabelas fato e de dimensões se combinam, em uma forma similar a um floco de neve.

O esquema Floco de Neve é bem similar ao esquema Estrela. No entanto, no Floco de Neve as dimensões são normalizadas com múltiplas tabelas que as representam. Já no esquema Estrela, cada dimensão é representada por uma única tabela.

Estruturas deste tipo são utilizadas em Data Warehouses, onde a velocidade de retorno de um dado é mais importante que a eficiência da manipulação dos mesmos. O esquema Floco de Neve é mais eficaz quando são utilizadas ferramentas de consulta mais sofisticadas, que criam uma camada de abstração entre o usuário e a estrutura propriamente dita.

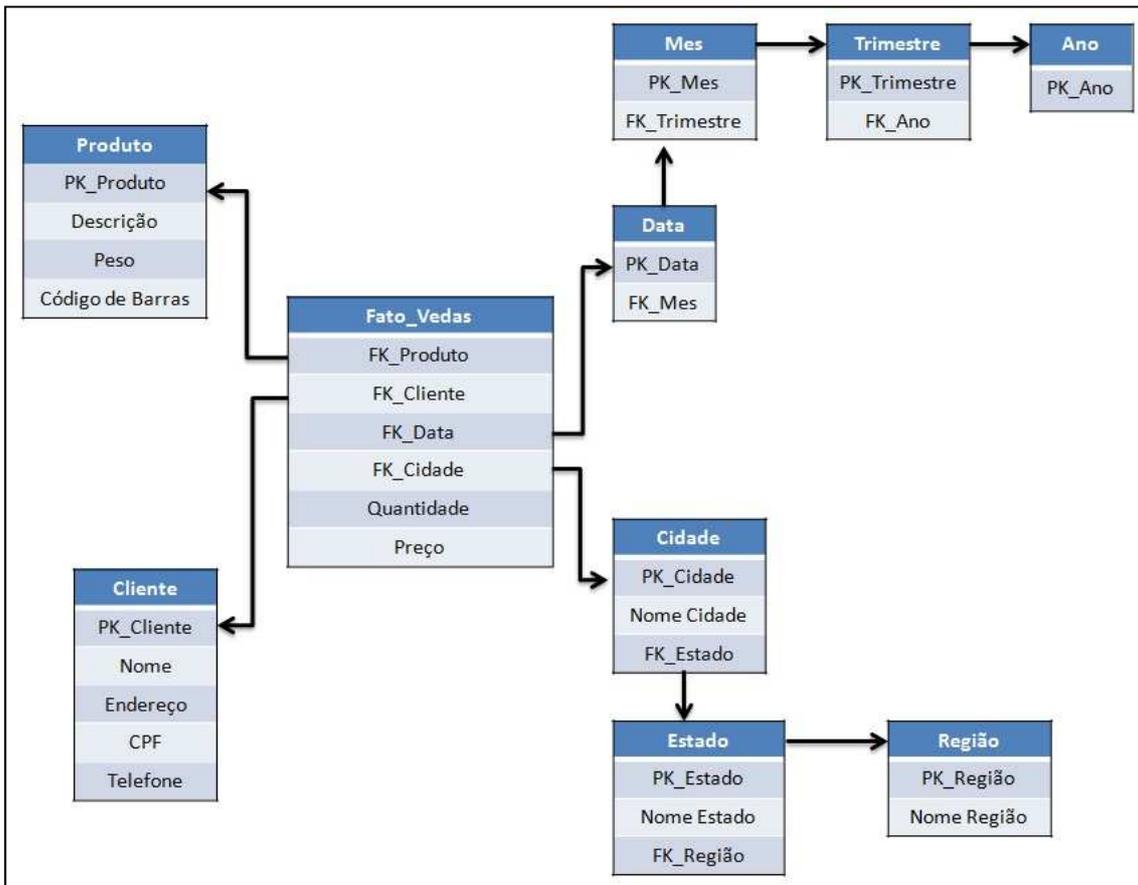


Figura 3: Modelagem de um esquema floco de neve

### 2.3.4 Esquema Estrela

Bastante similar ao esquema Floco de Neve, esta estrutura consiste em uma ou mais tabelas fato associadas às tabelas de dimensões. O número de dimensões associadas a uma tabela fato se limita à quantidade suficiente para satisfazer o entendimento da mesma.

A origem do seu nome é uma alusão a uma estrela, com uma tabela fato “central” e as dimensões em volta, representando as pontas da estrela.

Duas das principais vantagens desse esquema são o ganho de desempenho em operações de leitura comparado a outros esquemas normalizados e o cálculo de funções de agregação (soma, média, contagem) mais rápidas, de forma a resultar em um ganho significativo em operações de agregações.

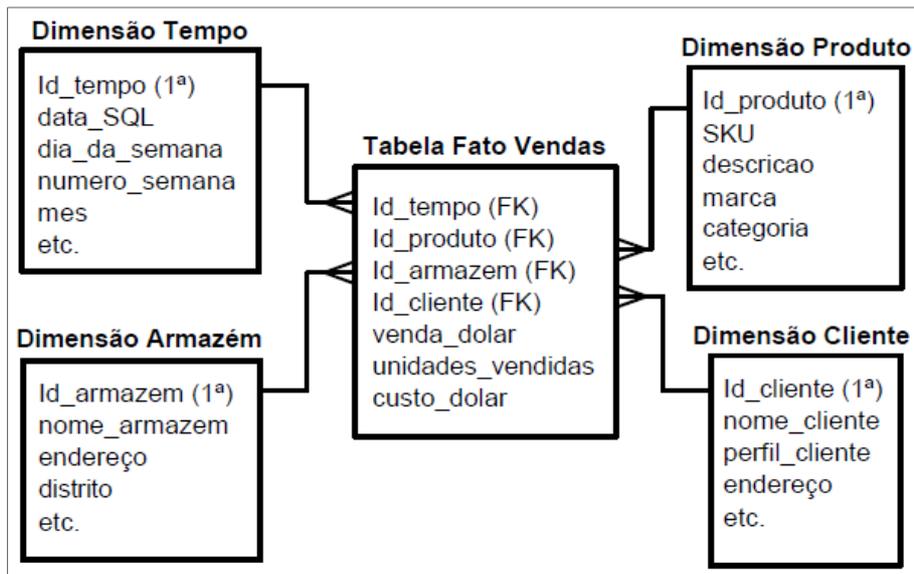


Figura 4: Modelagem de um esquema estrela

O esquema estrela é frequentemente referido como um cubo de dados, uma metáfora associada à tridimensionalidade de um cubo, embora o esquema estrela possa ter qualquer quantidade de dimensões. O uso do esquema estrela é uma prática comum no mercado, no desenvolvimento de Data Marts, que são componentes de um Data Warehouse corporativo, em geral associados a processos de negócio.

No presente trabalho, é utilizada a modelagem dimensional com esquema estrela, por ser reconhecidamente a técnica consagrada no mercado, em razão de suas vantagens em relação ao esquema dimensional floco de neve e à modelagem normalizada.

## 2.4 Definição das Métricas do Gartner Group para Avaliação de Plataformas de BI

Para a comparação entre as plataformas de BI Pentaho e OBIEE serão utilizadas as métricas definidas pelo Gartner Group. O Gartner Group é uma empresa de pesquisa e consultoria na área de Tecnologia de Informação. As pesquisas fornecidas pelo Gartner Group são direcionadas à CIOs e líderes seniores de TI em setores que incluem agências governamentais, empresas de alta tecnologia e de telecomunicações. O Gartner Group é bastante conhecido no mercado de TI e suas pesquisas são muito respeitadas e utilizadas em diversos contextos.

O Gartner Group estabelece métricas para avaliação de plataformas de BI, organizando-as em três categorias: integração, entrega de informação e análise.

### **Integração**

- Infraestrutura de BI: Todas as ferramentas na plataforma usam a mesma segurança, metadados, administração, integração de portal modelo de objeto e motor de consulta, e compartilham a mesma interface de usuário (*look and feel*).
- Gerenciamento de metadados: As ferramentas deve alavancar os mesmos metadados, e as devem proporcionar uma forma robusta para pesquisar, capturar, armazenar, reutilizar e publicar objetos de metadados, tais como dimensões, hierarquias, medidas, métricas de desempenho e objetos de layout do relatório.
- Ferramentas de desenvolvimento: a plataforma deve fornecer um conjunto de ferramentas de programação e visuais, juntamente com um kit de desenvolvimento de software para a criação de aplicativos analíticos, integrando-os em um processo de negócio e/ou incorporando-os em outro aplicativo.
- Colaboração: Permite aos usuários compartilhar e discutir informações e conteúdo analítico, e/ou gerenciar hierarquias e métricas através de tópicos de discussão, bate-papo e anotações.

### **Entrega de Informação**

- Relatórios: Provê habilidade para criar relatórios formatados e interativos, com sem parâmetros, com capacidades de distribuição e programação altamente escaláveis.
- Dashboards (Painéis): Inclui a capacidade de publicar relatórios com base na Web ou móveis com apresentações interativas intuitivas que indicam o estado de uma métrica de desempenho em comparação com um objetivo ou valor-alvo. Cada vez mais, os painéis são usados para divulgar dados em tempo real de aplicativos operacionais, ou em conjunto com um motor de processamento complexo de eventos.

- Consulta Ad hoc: Permite aos usuários fazer suas próprias consultas aos dados, sem depender de TI para criar um relatório. Em particular, as ferramentas devem ter uma camada semântica robusta para permitir que os usuários naveguem nas fontes de dados disponíveis.
- Integração com Microsoft Office: Às vezes, o Microsoft Office (principalmente Excel) atua como cliente de relatório ou análise. Nestes casos, é fundamental que a ferramenta proporcione integração com o Microsoft Office, incluindo suporte para formatos de documentos e apresentação, fórmulas, refresh de dados e tabelas pivot. A integração avançada inclui bloqueio de células e write-back.
- BI baseada em pesquisa: Permite aplicar um índice de pesquisa de fontes de dados estruturados e não estruturados e mapeia-os em uma estrutura de classificação de dimensões e medidas que os usuários podem facilmente navegar e explorar usando uma interface de busca.
- BI móvel: Permite que as organizações forneçam conteúdo analítico para dispositivos móveis em modo de publicação e/ou interativo, e tira proveito de reconhecimento de localização do cliente móvel.

## **Análise**

- Processamento analítico online (OLAP): Permite aos usuários analisar os dados com consulta rápida e desempenho no cálculo, permitindo um estilo de análise conhecida como "*slice and dice*". Os usuários são capazes de navegar caminhos de pesquisa multidimensional. Eles também têm a capacidade de escrever de volta valores em um banco de dados para propósito de modelagem e planejamento "*what if*". Esta capacidade pode abranger uma variedade de arquiteturas de dados (como relacional ou multidimensional) e arquiteturas de armazenamento (tais como baseado no disco ou na memória).
- Visualização interativa: dá aos usuários a capacidade de exibir vários aspectos dos dados mais eficientemente usando figuras e gráficos interativos, em vez de linhas e colunas.

- Modelagem preditiva e mineração de dados: permite às organizações classificar variáveis categóricas, e estimar as variáveis contínuas usando algoritmos matemáticos.
- Scorecards: Permite utilizar as métricas exibidas em um painel um pouco além, aplicando-as a um mapa estratégico que alinha os indicadores-chave de desempenho (KPIs) com um objetivo estratégico.

## 3 Descrição e Comparação das Plataformas OBIEE e Pentaho

### 3.1 Oracle Business Intelligence Enterprise Edition 11g

#### 3.1.1 Introdução

A suíte Oracle Business Intelligence Enterprise Edition 11g é uma plataforma de BI que possui diversas funcionalidades como *dashboards* interativos, consultas *ad hoc*, monitoramento das atividades do negócio e geração de alertas, relatórios empresariais e financeiros, *scorecard* e gerenciamento estratégico, pesquisa e colaboração.

A arquitetura do OBIEE é orientada a serviços, o que torna possível a integração com a infraestrutura de tecnologia de informação existente em uma organização.

#### 3.1.2 Arquitetura

Nesta seção será apresentada uma visão geral da arquitetura do OBIEE

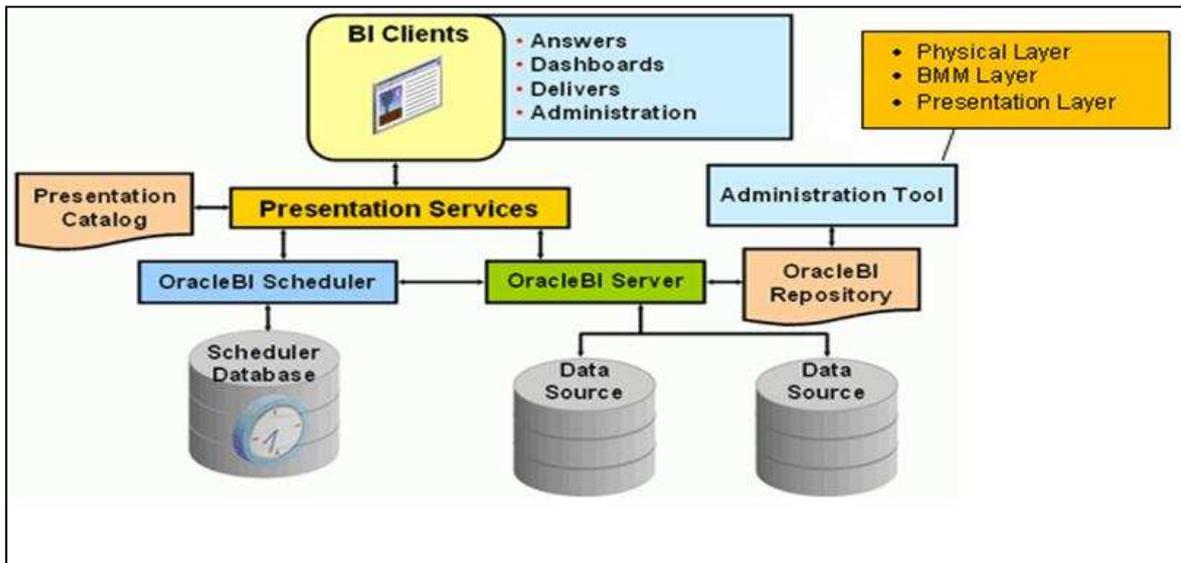


Figura 5: Diagrama da arquitetura do OBIEE <sup>11</sup>

### 3.1.2.1 BI Clients

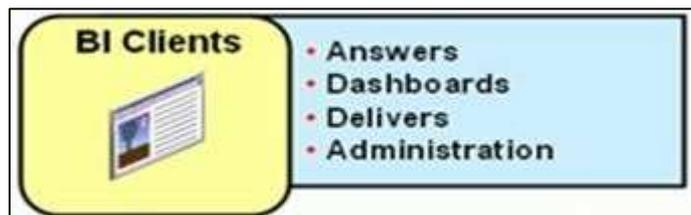


Figura 6: Diagrama da arquitetura do OBIEE - BI Clients

Este nível possui as aplicações Web utilizadas pelos usuários para a criação e visualização de artefatos como *dashboards*, consultas *ad hoc*, análises, relatórios e outros artefatos utilizados para a visualização dos dados do negócio. A seguir são apresentados os componentes do OBIEE que fazem parte desta área do ambiente.

#### Oracle Business Intelligence Dashboards

Esse componente permite a criação de *Dashboards* interativos. O acesso e navegação no Oracle BI Dashboards ocorre em ambiente Web. Usuários podem fazer *pivot*, *drill* e filtrar dados navegando no *Dashboard*. Além disso, possui também outras funcionalidades como *prompts*, navegação guiada e outras opções utilizando o botão direito do mouse.

Os usuários visualizam os dados e metadados de acordo com sua identidade, função ou papel baseado em configurações de segurança pré-definidas.

<sup>11</sup> <http://oracle-bi.siebelunleashed.com/articles/obiee-architecture/>

Permite a exportação para produtos Microsoft Office e adicionar *Dashboards* em uma pasta de favoritos. É possível a visualização em dispositivos móveis utilizando o Oracle BI Mobile.

### **Oracle Business Intelligence Answers**

Usuários finais do Oracle Business Intelligence Enterprise Edition podem criar análises *ad hoc* ou modificar análises contidas em *Dashboards* em um ambiente Web utilizando o componente Oracle BI Answers da suíte.

O usuário de negócio pode criar ou modificar uma análise *ad hoc* sem precisar conhecer a camada física do banco de dados. O OBIEE possui uma camada lógica onde é possível criar modelos lógicos do negócio a partir da camada física, permitindo maior clareza na visualização dos metadados e maior facilidade na geração de novas análises *ad hoc*.

### **Oracle Business Intelligence Delivers**

Oracle Business Intelligence Delivers tem como objetivo monitorar as atividades do negócio e gerar alertas que podem ser enviados para os usuários através de e-mail, *dashboards* e dispositivos móveis. A criação dos monitoramentos e alertas é feita em ambiente Web.

### **Oracle Business Intelligence Publisher**

Oracle BI Publisher é o produto utilizado para criação, gerenciamento e visualização de relatórios. É um produto separado do OBIEE, porém oferece integração com o mesmo e está incluso na compra desta suíte.

Este produto é usado para criação de relatórios operacionais, documentos de transações financeiras, formulários PDF, relatórios de marketing e outros documentos.

O usuário final pode criar análises *Ad hoc* e exportar os relatórios para o Microsoft Excel. Também é possível agendar o envio de relatórios para destinos como impressora, fax e e-mail.

O BI Publisher pode se conectar a diferentes fontes de dados como fontes relacionais, multidimensionais, *web service* e XML. Também pode utilizar dados sem estrutura do Oracle Endeca Server. Possui integração com o OBIEE permitindo transferência de dados e templates de dashboards e análises.

### 3.1.2.2 Presentation Services e Presentation Catalog

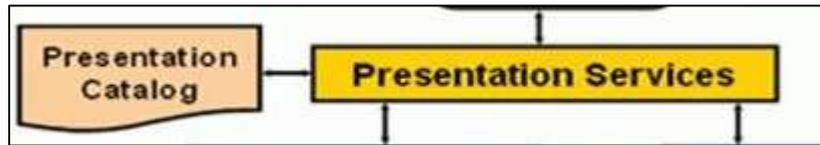


Figura 7: Diagrama da arquitetura do OBIEE - Presentation Services e Presentation Catalog

O Presentation Services possui um servidor Web onde aplicações BI como Oracle BI Answers e Oracle BI Dashboard são executadas. As requisições dos usuários são processadas e são geradas consultas SQL lógicas baseadas na estrutura do *Dashboard*, análise *ad hoc* ou relatório visualizado. Essa consulta é enviada para o Oracle BI Server que gera uma consulta SQL física.

Uma consulta SQL lógica é uma consulta SQL padrão, que suporta algumas funções extras, executada sobre a camada de apresentação de um repositório do OBIEE. Uma consulta SQL física é uma consulta executada pelo Oracle BI Server sobre a camada física de um repositório do OBIEE. Além dessas duas camadas, existe uma camada intermediária chamada de camada de modelo de negócios. Essas três camadas e o repositório serão explicados em detalhes no tópico Oracle BI Server, Administration Tool e Oracle BI Repository.

O Presentation Catalog armazena os artefatos criados pelos usuários como *dashboards*, análises, relatórios, pastas e filtros. Também possui informações sobre as permissões que os usuários têm sobre os artefatos. O Presentation Catalog é administrado através da ferramenta Catalog Manager. Esta é uma ferramenta *Desktop*.

Em resumo, o Presentation Services e o Presentation Catalog fornecem juntos um servidor web onde as aplicações web são executadas e também administram como a interface do usuário será apresentada.

### 3.1.2.3 Oracle BI Scheduler e Scheduler Database



Figura 8: Diagrama da arquitetura do OBIEE - Oracle BI Scheduler e Scheduler Database

O Oracle BI Scheduler possui um servidor chamado Oracle BI Scheduler Server que gerencia e realiza o agendamento de *jobs* do Oracle BI Scheduler.

Existem dois tipos de *jobs*.

**Agentes.** Agentes entregam conteúdo para usuários finais. O conteúdo pode ser análises, *dashboards*, relatórios ou alertas. Após entregar o conteúdo, agentes também podem executar ações. Essas ações só podem ser executadas após a entrega do conteúdo. Ações podem executar um programa em Java, acessar uma URL, iniciar um *Web Service* e executar *scripts* de servidor. Agentes podem executar outros agentes, criando uma cadeia de execução de agentes. Agentes são configurados e submetidos para execução utilizando o Oracle BI Delivers.

**Scripted Jobs.** Esses *jobs* são configurados e submetidos utilizando o gerenciador de *jobs* presente no Oracle BI Administration Tool. Por exemplo, um *Scripted Job* poderia inserir as estatísticas de uso do BI Server em um banco de dados ou salvar a saída de um agente do Oracle BI Delivers em uma pasta compartilhada do sistema operacional. *Scripted Jobs* podem ser escritos em Java, VBScript e JScript.

#### 3.1.2.4 Oracle BI Server, Oracle BI Repository e Administration Tool

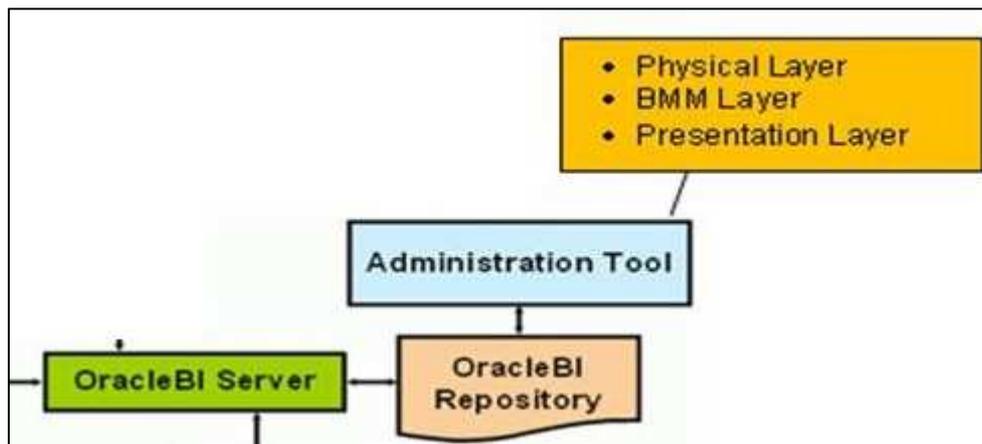


Figura 9: Diagrama da arquitetura do OBIEE - BI Server, Admin Tool e BI Repository

### Oracle BI Server

O Oracle Business Intelligence Server é a fundação da suíte OBIEE. Tem como objetivo centralizar o acesso às diferentes fontes de dados de uma empresa, bem como cálculos e agregações.

O BI Server é o centro de informações para *dashboards*, consultas *ad hoc*, análises, relatórios empresariais e de produção, relatórios financeiros, análises OLAP, mineração de dados, e outras aplicações baseadas em Web Service como J2EE e .NET.

O BI Server não armazena dados. Armazena apenas os metadados das fontes de dados para gerar e executar consultas sobre essas fontes de dados. Um de seus principais objetivos é integrar de maneira eficiente as diversas fontes de dados de uma empresa, sejam elas relacionais, multidimensionais ou desestruturadas.

O BI Server recebe as consultas SQL lógicas feitas sobre a camada de apresentação do repositório, interpreta essas consultas e gera as consultas físicas, que são executadas sobre a camada física do repositório.

### Oracle BI Repository

O Oracle BI Repository é um arquivo que contém todos os metadados da camada de apresentação, da camada de modelo de negócio e da camada física de um repositório. Um repositório é criado utilizando o software para Windows chamado Oracle BI Administration Tool. A figura 10 mostra a interface do Administration Tool.

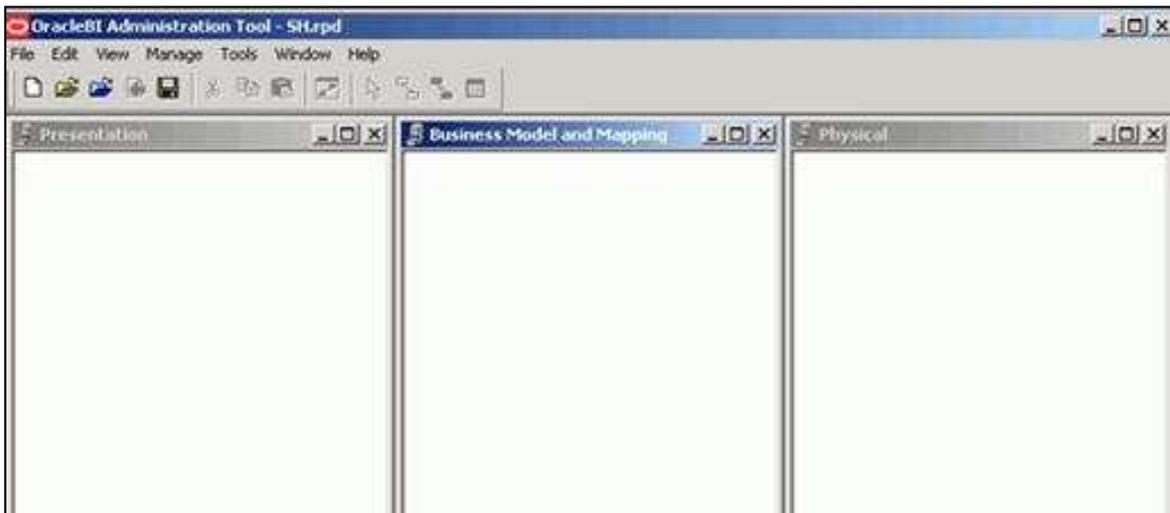


Figura 10: Interface do Administration Tool

Um repositório possui três camadas.

**Camada Física.** Essa camada possui os objetos necessários para que o Oracle BI Server gere as consultas nativas que serão executadas sobre a fonte de dados física. As consultas físicas podem ser geradas em diferentes linguagens como, por exemplo, linguagem SQL para consultas em bancos relacionais, e linguagem MDX para consultas em cubos OLAP.

A camada física é criada importando tabelas, cubos e arquivos externos das fontes de dados.

Com a separação da camada física da camada lógica, é possível utilizar diferentes fontes de dados físicas para um mesmo objeto lógico, possibilitando navegação em dados agregados e particionados, como também isolamento de mudanças nas fontes de dados físicas.

**Camada de Modelo de Negócio e Mapeamento.** Nessa camada é definido o modelo de negócio (ou modelo lógico) dos dados e é feito o mapeamento entre um modelo de negócio e suas fontes de dados físicas. O mapeamento não é feito automaticamente. Essa camada determina como será a visualização analítica dos usuários e define o conjunto de objetos e relacionamentos que os usuários poderão ter acesso. Além disso, também esconde a complexidade da camada física. A camada de modelo de negócios e mapeamento é obrigatoriamente dimensional.

Cada tabela no modelo de negócios é mapeada para um ou mais objetos da camada física. Em tempo de execução, o Oracle BI Server analisa a consulta SQL lógica gerada

sobre o modelo de negócios, e então usa os mapeamentos para determinar o melhor conjunto de tabelas, arquivos externos e cubos para gerar a consulta física. Os mapeamentos normalmente contêm cálculos e agregações.

**Camada de Apresentação.** Essa camada permite criar diferentes pontos de vista sobre um modelo de negócio de maneira customizada, com segurança e acesso aos dados baseado em função de usuário. Adiciona uma camada de abstração sobre a camada de modelo de negócios e mapeamento e determina como os dados e metadados serão visualizados pelos usuários quando estes acessarem *dashboards*, análises *ad hoc* e relatórios.

Cada ponto de vista é chamado de área de assunto. É possível criar diferentes áreas de assunto utilizando um único modelo de negócio, dividindo o modelo de negócio em partes gerenciáveis.

A seguir, a figura 11 mostra uma visão geral das três camadas de um repositório e o fluxo de dados de uma consulta gerada pelo usuário final.

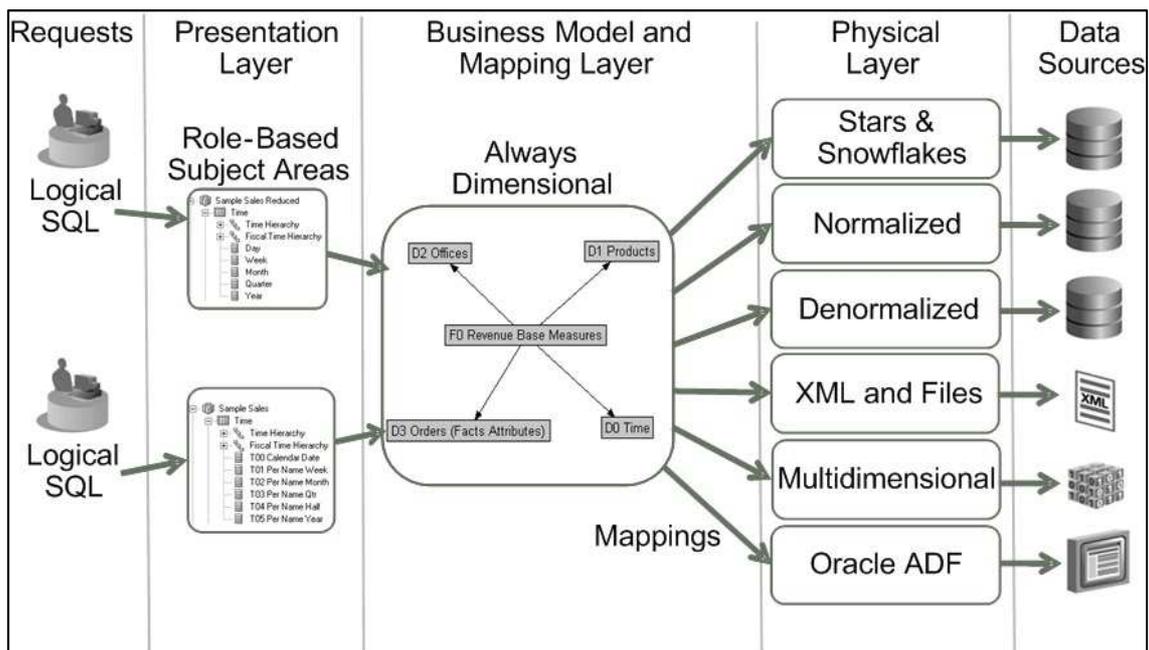


Figura 11: Camadas de um repositório e fluxo de dados gerado por uma consulta<sup>12</sup>

### 3.1.2.5 Datasources

<sup>12</sup> [http://docs.oracle.com/cd/E23943\\_01/bi.1111/e10540/intro.htm#CIHEDEGE](http://docs.oracle.com/cd/E23943_01/bi.1111/e10540/intro.htm#CIHEDEGE)

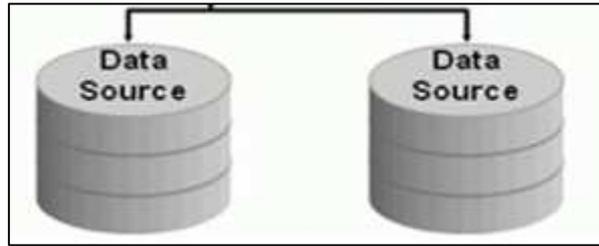


Figura 12: Diagrama da arquitetura do OBIEE - Datasources

Os *dashboards*, análises e relatórios criados no OBIEE utilizam dados armazenados em bancos de dados subjacentes. Ou seja, o OBIEE não possui um banco de dados para armazenar os dados utilizados por *dashboards* e análises. Na verdade, o OBIEE faz consultas nos bancos de dados subjacentes utilizando os metadados desses bancos de dados que foram importados para a camada física de um repositório do OBIEE. As consultas são geradas pelo Oracle BI Server utilizando os metadados importados na camada física.

*Dashboards* e análises podem gerar consultas que necessitam de dados de mais de um banco de dados da camada física. O Oracle BI Server possui a capacidade de executar consultas em diferentes bancos de dados para gerar o conjunto de dados necessário para a consulta requisitada pelo usuário final.

O OBIEE suporta os seguintes tipos de conexão: ODBC 2.0, ODBC 3.5, DB2 CLI (Unicode), XML, Essbase 9, XMLA, OracleADF\_HTTP, Oracle OLAP, OCI 10g/11g, Hyperion ADM e BW de SAP Nativo.

### 3.1.2.6 Visão geral do processo de consulta aos dados

O processo de consulta aos dados ocorre da seguinte forma: quando um usuário requisita uma consulta através de, por exemplo, *dashboards* ou análises *ad hoc*, a requisição primeiramente segue para o Presentation Services e, em seguida, é roteada para o BI Server. Depois, a requisição segue para a fonte de dados (banco relacional, multidimensional, etc.).

A resposta para essa requisição segue o seguinte fluxo: os dados são obtidos de uma ou mais fontes de dados pelo BI Server. Depois, a resposta é recebida pelo Presentation Services que, por fim, envia os dados para o Client (dashboard, análise ad hoc).

## 3.2 Pentaho BI Suite

### 3.2.1 Introdução

Há quase 10 anos no mercado, o Pentaho é uma suíte completa <sup>13</sup>de BI resultante do empacotamento de diversas ferramentas de BI desenvolvidas independentemente como software livre. Diferentemente do OBIEE e de outras disponíveis, o Pentaho é *open source*, isto é, possui licença livre para uso e o código é aberto para desenvolvimento cooperativo pela comunidade. Além disso, conta com um suporte *online* para resolução de possíveis questionamentos durante o desenvolvimento do projeto.

Nesta seção, serão apresentados os módulos em que esta suíte está dividida. Estes módulos provêm todas as ferramentas necessárias para o desenvolvimento de um projeto de Business Intelligence, desde a extração dos dados até a montagem de relatórios para os usuários finais.

### 3.2.2 Community Edition x Enterprise Edition

#### Community Edition

O Pentaho possui duas edições: Community Edition e Enterprise Edition. Para elaboração deste projeto foi utilizada a edição para comunidade (Community Edition), que é *open source* e está disponível gratuitamente para *download* no sítio do Pentaho.

---

<sup>13</sup> <http://wiki.pentaho.com/display/COM/Community+Wiki+Home>

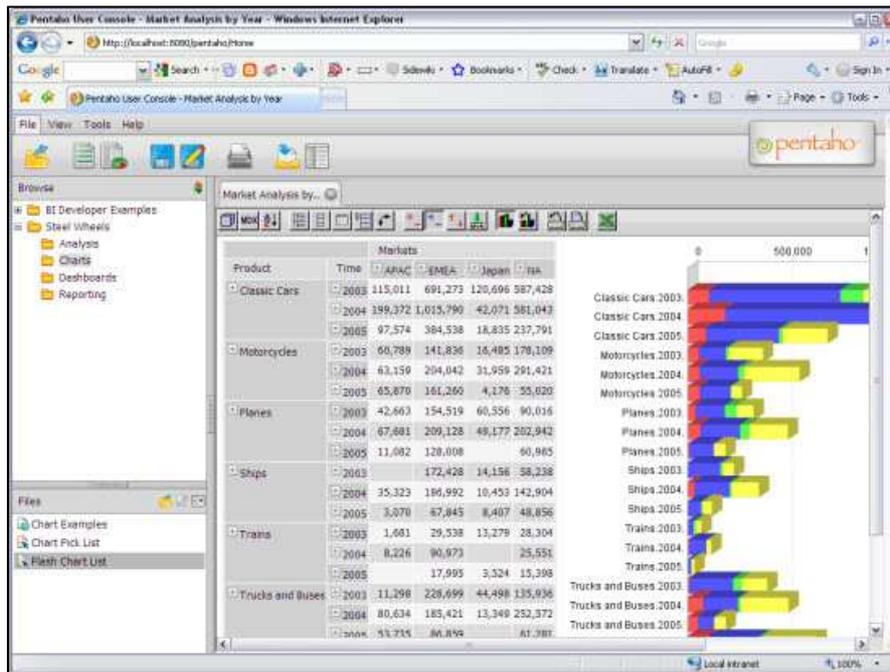


Figura 13: Relatório no Pentaho Community Edition

### Enterprise Edition

Em contrapartida, a edição para empresas (Enterprise Edition) possui uma versão de avaliação por 30 dias e, após esse período, é necessário comprar a licença para utilização, pois trata-se de um software comercial.

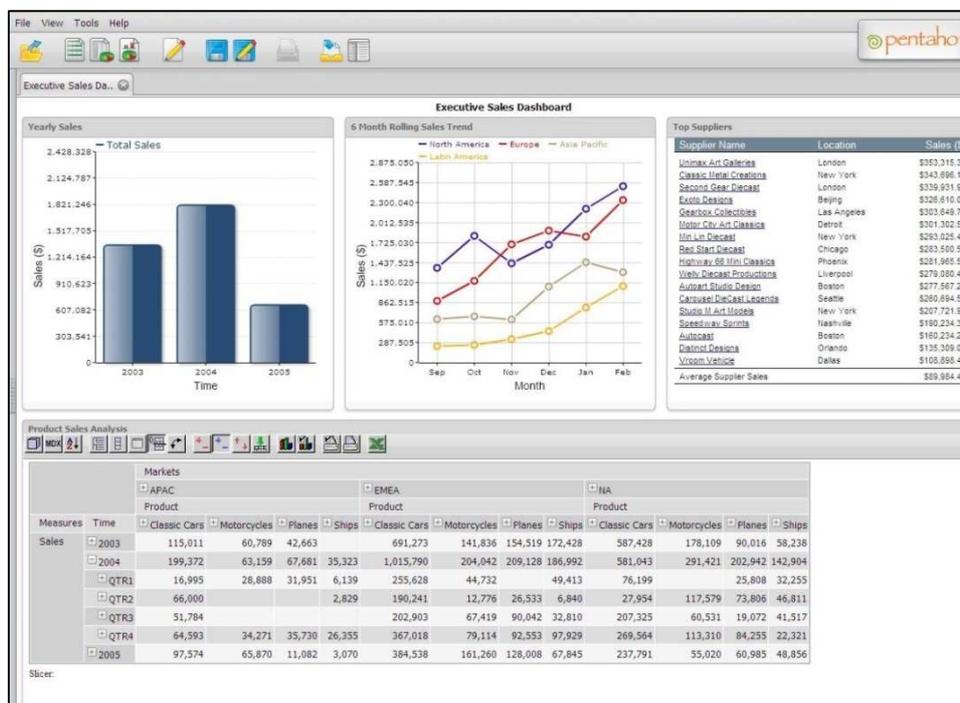


Figura 14: Relatório no Pentaho Enterprise Edition

As principais e mais notáveis diferenças entre as versões do Pentaho são o preço e as atualizações. Na versão empresarial, embora seja preciso pagar, o produto é constantemente atualizado. Além disso, conta com um suporte bem prestativo e uma interface de assistente para instalação e configuração, que são ferramentas que não estão presentes na versão Community.

### 3.2.3 Pentaho BI Platform

O Pentaho BI Platform consiste em um servidor de BI, além de alguns clientes e *add-ons* que serão citados posteriormente. Esta plataforma oferece algumas funcionalidades, entre elas:

- Um console do usuário na Web, que torna mais fácil o desenvolvimento e o gerenciamento das análises e relatórios.
- Uma interface *ad-hoc* de relatórios, que oferece um guia passo-a-passo para montagem de relatórios simples e em diversos formatos (PDF, RTF, HTML e XLS, por exemplo).
- Interface de visualização de análises em tempo real, onde é permitido realizar *drill-down* em dados preparados para tal.
- Um complexo sistema de agendamento que permite aos usuários definir em qual momento um determinado relatório deve ser executado.
- A possibilidade de publicar relatórios por e-mail para outros usuários.
- Conectividade total com o Pentaho Metadata e Pentaho Report Designer, para uma maior facilidade de criação de conteúdos.

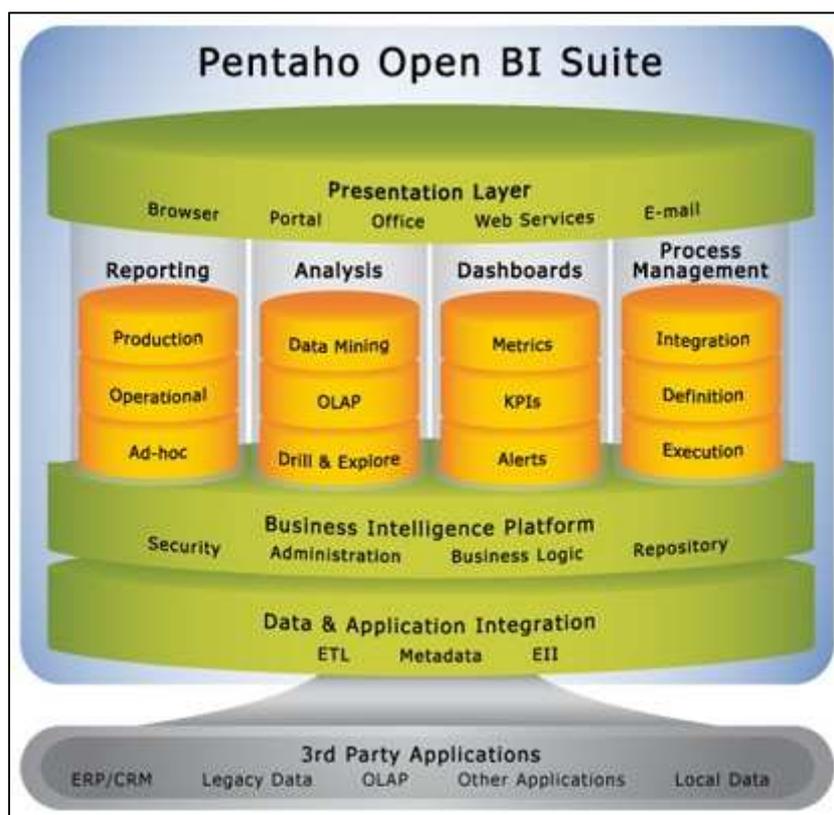


Figura 15: Representação da arquitetura do Pentaho

### 3.2.4 Pentaho Metadata Editor (PME)

É uma ferramenta que constrói os modelos e domínios de metadados do Pentaho. Um modelo de metadados no Pentaho mapeia a estrutura física da base de dados em um modelo de negócios lógico. Esses mapeamentos são armazenados em um repositório de metadados e permitem as seguintes ações:

- Criação de uma linguagem com definições do negócio para tabelas complexas do banco de dados.
- Diminuição do custo e impacto associado às mudanças de baixo nível em um SGBD.
- Definição de parâmetros de segurança, limitando o acesso de usuários a dados de relatórios.
- Localização das informações de acordo com as definições regionais dos usuários.

O modelo de metadados de negócio é o principal componente do domínio de metadados do Pentaho. O domínio encapsula tanto as descrições dos objetos da base de

dados física quanto o modelo lógico (modelo de negócios), que é a representação abstrata do banco.

### **Camada Física**

A camada física de um domínio de metadados no Pentaho é constituída pelas tabelas físicas e colunas físicas. Estes objetos representam o banco de dados que está tentando se modelar e enriquecer com metadados.

Esta camada não faz parte do modelo de negócios, pois nem todos os objetos definidos nela serão utilizados.

### **Camada Abstrata de Negócio**

Esta camada é a parte mais complicada do modelo de metadados de negócio. Nesta camada estão presentes as tabelas, colunas e relacionamentos, ambas referentes ao negócio.

Pode-se criar tabelas de negócio para qualquer tabela física que está definida na camada física. Também se pode criar mais de uma tabela de negócio referenciando uma mesma tabela física. As mesmas regras se aplicam às colunas. Isto pode ser útil para diversos cenários, como, por exemplo, definindo níveis de segurança de acesso.

A tabela de negócio guarda uma referencia à tabela física, o que permite uma relação de herança entre as mesmas. Por exemplo, se é definido um metadado para uma tabela física, a tabela de negócio irá herdar este metadado, até que esta tabela de negócio substitua o metadado que foi herdado. O mesmo vale para as colunas.

### **A Visão de Negócio**

É a parte do modelo de negócio que os usuários finais irão visualizar. As visões do negócio são, basicamente, categorias onde é possível reorganizar as colunas em um layout que faça sentido para os consumidores dos dados.

É permitida a criação de quantas categorias forem necessárias, além de organizar as colunas das tabelas para a visão que melhor se adequar ao negócio.

## **3.2.5 Pentaho Design Studio (PDS)**

É uma coleção de editores, visualizadores e módulos de administração integrados em uma única aplicação que fornece um ambiente gráfico para montagem e testes de

ações, que podem ser atividades como consultas ao banco, geração de relatórios e envio de e-mails. O PDS auxilia na definição da ordem em que essas ações irão ocorrer.

### **3.2.6 Pentaho Administration Console (PAC)**

Nesta ferramenta é onde está centralizada a administração do desenvolvimento do projeto. O PAC agrega e simplifica as mais comuns tarefas administrativas, como o gerenciamento de usuários e papéis, agendamento de tarefas e gerenciamento de serviços. Este console permite uma melhor interação com o desenvolvimento do projeto, automatizando algumas tarefas que são normalmente executadas manualmente.

O Pentaho Administration Console oferece funcionalidades limitadas, se comparado ao Pentaho Enterprise Console. O segundo é capaz de fornecer funcionalidades adicionais que permitem monitorar performance, testar configurações, configurar a segurança e muito mais.

### **3.2.7 Pentaho Data Integration (PDI)**

O Pentaho Data Integration – PDI, originalmente denominada de Kettle, é uma ferramenta livre *open-source* responsável por fornecer um ambiente capaz de realizar ETL de diversos tipos de dados.

#### **ETL (Extract, Transform and Load)**

Por ser uma ferramenta de ETL, o Kettle é um ambiente projetado para:

- Coletar dados de múltiplas fontes (*extract*).
- Mover e alterar dados (*transport and transform*) enquanto limpa, desnormaliza, agrega e enriquece os mesmos no processo.
- Frequentemente (dependendo da periodicidade, mas normalmente diariamente) armazenar dados (*loading*) no seu destino final que é, normalmente, um DW modelado dimensionalmente.

#### **Arquitetura do PDI**

Esta ferramenta consiste de quatro diferentes aplicações. São elas:

- Spoon → Ferramenta que permite modelar o fluxo de dados desde a entrada de dados (*input*), passando pela transformação e gerando a saída (*output*).
- Pan → Ferramenta em linha de comando que executa transformações modeladas com o Spoon.
- Chef → Ferramenta utilizada para modelar *jobs*. *Jobs* são tarefas como transformações, download de arquivos por FTP e outras atividades que compõem um fluxo de controle.
- Kitchen → Ferramenta em linha de comando que executa *jobs* criados com o Chef.

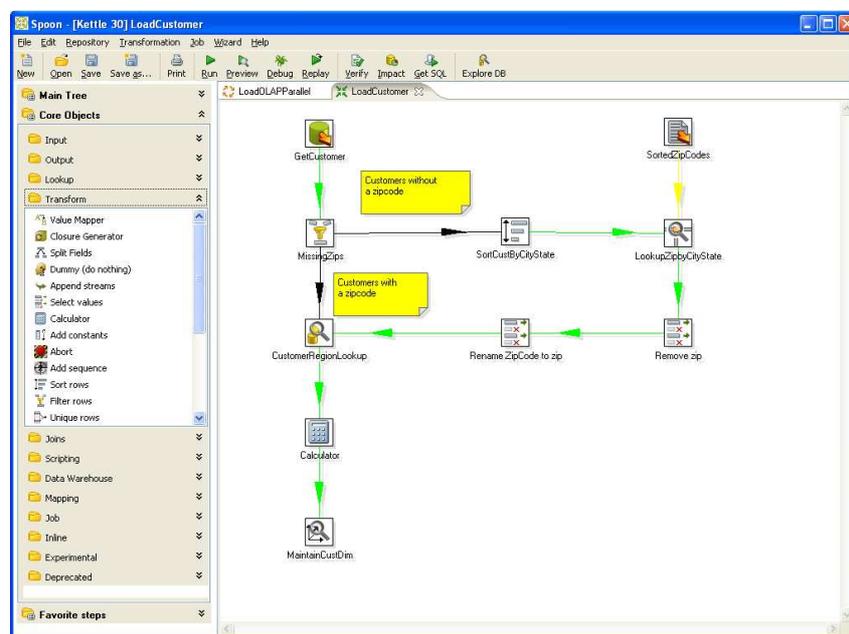


Figura 16: Representação de uma transformação no Kettle

### 3.2.8 Pentaho Analysis Services (Mondrian OLAP Server)

O Mondrian é uma ferramenta OLAP escrita em Java que executa consultas criadas em linguagem MDX, lendo dados de uma base de dados relacional e apresentando resultados em um formato multidimensional através de uma API Java. Ele está dividido em quatro camadas:

#### Camada de Apresentação

A camada de apresentação determina o que o usuário final vê e como ele pode interagir para levantar novos questionamentos de negócio. Existem diversas formas para apresentação de conjuntos de dados multidimensionais, como os gráficos em pizza,

linha, barra e também algumas ferramentas de visualização avançada como os gráficos dinâmicos e mapas clicáveis.

### Camada Dimensional

Esta camada permite analisar, validar e executar consultas MDX. Uma consulta é avaliada em múltiplas fases e existe um transformador de consultas que permite que a aplicação manipule consultas já existentes, em vez de construir uma nova consulta para cada nova requisição.

### Camada Estrela

Responsável por manter dados agregados. Uma agregação é um conjunto de métricas em memória, qualificado por um conjunto de colunas de dimensão. A camada dimensional envia requisições para a camada estrela e, caso as tuplas não estejam na camada estrela, o gerenciador de agregações envia uma requisição para a camada de armazenamento.

### Camada de Armazenamento

Nada mais é que o banco de dados relacional, que é responsável por fornecer dados agregados e os membros das tabelas de dimensão.

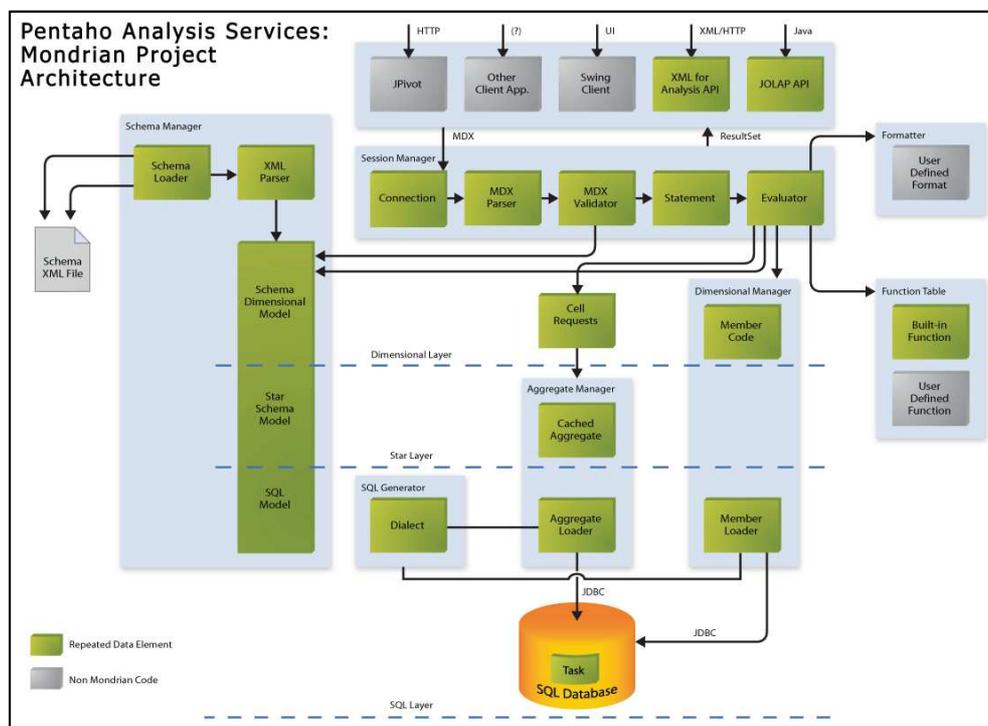


Figura 17: Arquitetura do Pentaho Analysis Services

### 3.2.9 Pentaho Community Dashboard Editor (CDE)

Esta ferramenta é integrada com o Pentaho User Console e possibilita a criação, edição e visualização de *dashboards*. Para o projeto de *dashboards*, o Pentaho CDE oferece 3 perspectivas:

### **Layout**

Projeta o layout do *dashboard* a partir de um rascunho ou de um *template*. Enquanto o layout está sendo definido pode-se aplicar estilos diferentes e adicionar elementos HTML como textos ou imagens.

### **Componentes**

Adiciona e configura os diferentes componentes que constroem o *dashboard*: Componentes visuais (caixas de texto, tabelas, gráficos, etc.), parâmetros e *scripts*.

### **Fontes de dados**

Define as fontes de dados utilizadas pelos componentes definidos nos itens acima.

Além disso, o CDE conta com uma opção de pré-visualização, que permite testar a visão e o comportamento do *dashboard* enquanto ele está sendo desenvolvido.

## **3.2.10 Pentaho Aggregation Designer (PAD)**

Simplifica a criação e o desenvolvimento de tabelas agregadas que melhoram a performance dos cubos OLAP criados no Mondrian.

O Pentaho Analysis é um mecanismo relacional OLAP que trabalha unicamente com os dados armazenados na base de dados relacional, em vez de fornecer seu próprio modelo de armazenamento multidimensional. Isso simplifica o desenvolvimento e o gerenciamento dos dados, mas limita a performance quando são trabalhados grandes volumes de dados (tabelas fato com mais de 10 milhões de registros e/ou cubos com cardinalidade alta de níveis e membros). Para melhorar a performance nestes cenários, o Pentaho Analysis suporta tabelas agregadas.

O PAD fornece uma interface simples que permite criar tabelas agregadas de níveis com dimensões especificadas pelo desenvolvedor. Baseado nestas seleções, o Aggregation Designer gera a DDL para criação de tabelas agregadas, a DML para a manipulação das mesmas e um esquema Mondrian atualizado que referencia as novas tabelas agregadas.

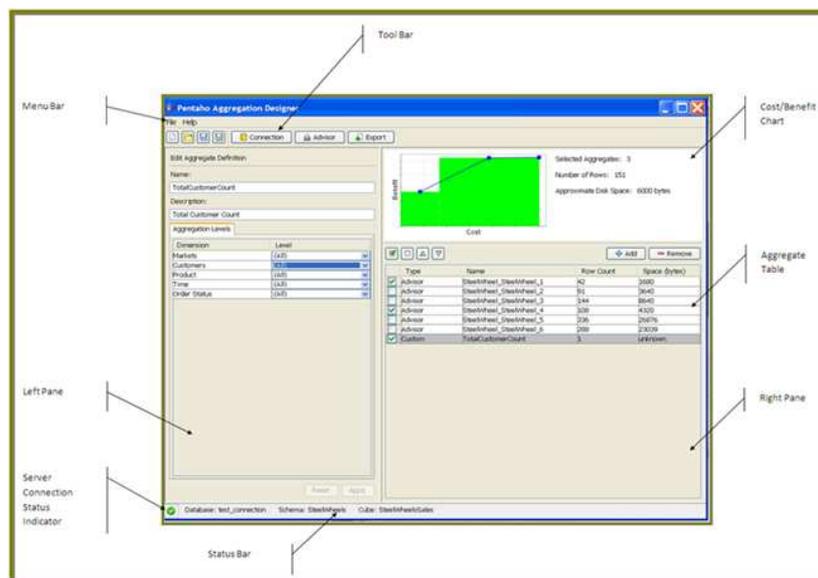


Figura 18 - Interface do Aggregation Designer

### 3.2.11 Saiku Analytics

O Saiku Analytics é um cliente web disponível como *plug-in* para o Pentaho BI Server. Ele usa o Pentaho Analysis (Mondrian) para proporcionar uma forma fácil de usar o recurso de Cubos OLAP com uma experiência simples para usuário final.

Neste trabalho, foram criadas análises utilizando o Saiku e foi percebido que a experiência no desenvolvimento é muito melhor ao utilizá-lo. A montagem das análises é imediata e menos trabalhosa que na ferramenta que já vem disponível no BI Server.

## 3.3 Aplicação das Métricas no Pentaho e no OBIEE

Nesta seção, as duas plataformas de BI são comparadas utilizando as métricas definidas pelo Gartner Group. Não se trata de uma comparação de software para seleção de fornecedor e aquisição utilizando um passo-a-passo como, por exemplo, o apresentado no artigo "Passo-a-Passo para Avaliação de Software para Aquisição em Modelos de Processos de Negócio", Azevedo, Romeiro, Cappelli e Baião (2010). Nesta seção, é realizada apenas uma comparação geral entre as duas plataformas e, no capítulo 7, são comparadas características de desenvolvimento de uma aplicação analítica usando essas plataformas.

### 3.3.1 Integração

#### Infraestrutura

Uma premissa básica do Gartner para a avaliação em termos de infraestrutura é a capacidade de desenvolvimento de todas as funcionalidades de BI em uma única instalação. Tomando isso como base, avaliamos negativamente o Pentaho pois é necessária a instalação de diversos componentes para poder usufruir de suas funcionalidades. Enquanto o motor de consulta é único para os componentes, a segurança e gerenciamento de metadados podem ser feitos em plataformas diferentes. Já a administração de papéis e usuários também é feita em outra interface.

Além disso, existem limitações na Community Edition. Funcionalidades como instaladores automáticos, automatização de *jobs* ETL e integração com Big Data só estão disponíveis na versão empresarial.

No OBIEE, existe boa integração entre as ferramentas que compõem a plataforma. As consultas requisitadas em relatórios e análises são geradas basicamente por um único motor de consultas, o BI Server, que acessa um único repositório de metadados do OBIEE.

Utilizando uma interface gráfica na web é possível, por exemplo, visualizar logs, executar diagnósticos, configurar caches, implantar repositórios, e iniciar e parar serviços.

Há divisão entre a administração de metadados do repositório do OBIEE e a parte de *front end* do ambiente (criação de relatórios e análises). A administração dos metadados é feita utilizando um programa *desktop* para Windows e a administração do *front end* é feita através do portal web do OBIEE.

A administração da segurança de usuários, grupos e papéis é realizada utilizando três portais web diferentes e um programa para Windows.

### **Gerenciamento de metadados**

O gerenciamento de metadados no Pentaho é um tanto quanto complexo. Para os metadados de relatório, a criação, edição e segurança destes são feitas em uma única ferramenta. Já para a criação das dimensões, hierarquias e métricas dos cubos criados é utilizado outro componente. Embora desenvolvidos em ambientes diferentes, a publicação de cada um pela sua respectiva ferramenta é feita em um repositório único.

No OBIEE, o desenvolvimento e gerenciamento de metadados como dimensões, hierarquias e medidas são realizados através do programa *desktop* para Windows

chamado Administration Tool. Neste programa são realizadas as tarefas que definem o conjunto de metadados disponível para o usuário final. Os relatórios e análises utilizam este único conjunto de metadados.

O desenvolvimento e gerenciamento de objetos de *front end* como análises e relatórios são feitos através do portal web do OBIEE.

### **Ferramentas de desenvolvimento**

Segundo o Gartner, a avaliação deve ser feita levando em consideração a capacidade da ferramenta prover interfaces gráficas (*drag and drop, wizards*) para o desenvolvimento. Além disso, deve fornecer uma interface web para atividades comuns como manutenção e administração.

De forma geral, o Pentaho possui ferramentas de desenvolvimento de ponta a ponta para uma solução BI: desde a ETL até a criação de relatórios e *dashboards*. Todas estas ferramentas possuem interface gráfica, algumas menos amigáveis que outras e com algumas limitações no desenvolvimento, sendo necessário criar códigos. Por exemplo, no Mondrian (ferramenta OLAP do Pentaho BI Suite), é preciso editar arquivos XML para definição de parâmetros de segurança.

O OBIEE possui duas ferramentas principais para o desenvolvimento de aplicações BI. Para o desenvolvimento de dimensões e hierarquias é utilizado o Administration Tool e para o desenvolvimento de análises e relatórios é utilizado o portal web do OBIEE. As duas ferramentas do OBIEE oferecem um grande conjunto de funcionalidades para facilitar o desenvolvimento da aplicação.

### **Colaboração**

Não existe nenhuma forma de colaboração entre os usuários do Pentaho. Os relatórios e análises uma vez montados não são disponibilizados em pastas públicas e não existe nenhum processo definido para cada *output* gerado. Cada relatório é de propriedade exclusiva do seu autor.

No OBIEE é possível compartilhar relatórios e análises com outros usuários através de pastas públicas, porém esta é a única forma de colaboração.

## **3.3.2 Entrega de Informação**

### **Relatórios**

O Pentaho é altamente capacitado a criar relatórios com as mais diversas formatações. Além disso, também é possível, por exemplo, agendar ações para que um relatório específico seja enviado por e-mail em um determinado horário do dia. Através da sua ferramenta de elaboração de relatórios, o Pentaho permite a exportação em PDF, XLS e em outros formatos. A ferramenta também conta com *wizards* que auxiliam em todo o processo de criação do relatório, desde a conexão com a fonte de dados, passando pelas informações que desejam ser visualizadas até chegar na forma de exportação do relatório em si.

O OBIEE utiliza o Oracle BI Publisher para a criação de relatórios empresariais formatados. Esta ferramenta é, na verdade, um produto separado do OBIEE, porém está incluso na plataforma e possui excelente integração com a mesma. O BI Publisher possui diversas opções para criação e formatação do relatório, permite o envio agendado dos relatórios para diferentes destinos em diversos formatos. É possível interagir com os relatórios gerando novas consultas.

### ***Dashboards***

Apesar de criados através de um *plug-in* contido em uma de suas ferramentas, os *dashboards* do Pentaho são bem completos e acrescentam muito à tomada de decisão. Existem dois *plug-ins* que auxiliam na criação de *dashboards* no Pentaho: o Pentaho Community Dashboard Editor e o Pentaho Community Dashboard Framework. Os dois podem ser adquiridos diretamente no *marketplace* do Pentaho BI Server e a instalação é feita no próprio browser, sem a necessidade de download de um instalador à parte.

O OBIEE utiliza a ferramenta Oracle BI Dashboards para a criação de *dashboards*. A criação ocorre em ambiente web utilizando uma interface gráfica. É possível enviar os *dashboards* para diferentes destinos como e-mail e impressora, e em diferentes formatos como PowerPoint ou Excel. Para a visualização em dispositivos móveis, é necessário comprar outro produto da Oracle chamado Oracle BI Mobile.

### ***Consultas ad-hoc***

O Pentaho possui a capacidade de realizar consultas *ad-hoc* através de uma ferramenta denominada WAQR (Web Adhoc Query and Reporting), uma ferramenta de geração de relatório que fornece uma interface que ajuda o usuário a identificar e agrupar os dados que lhe interessam, aplicar restrições a eles e gerar um relatório.

O OBIEE utiliza o Oracle BI Answers para a geração de consultas *Ad-hoc*. Esta ferramenta permite que o usuário final crie suas próprias análises sem precisar conhecer a camada física do banco de dados. O usuário final pode criar ou modificar análises para atender as suas necessidades sem precisar realizar um pedido para a área de TI da empresa. Esta ferramenta disponibiliza uma grande quantidade de funcionalidades ao usuário final, porém sua interface gráfica pode ser confusa e o aprendizado da ferramenta pode ser um pouco mais demorado quando comparado com o Pentaho.

### **Integração com o Microsoft Office**

Embora não possua integração com o MS Office, existem diversos projetos em desenvolvimento <sup>14</sup> pela comunidade Pentaho que visam integrar o Pentaho BI Platform com softwares *open source* como o OpenOffice. Inclusive, há também uma ideia de integração com o MS Project.

O OBIEE utiliza a ferramenta Oracle Hyperion Smart View para integrar o OBIEE às ferramentas Microsoft Office. O Smart View é um *plug-in* para as ferramentas Office, como Word e Excel, que permite, por exemplo, visualizar análises e relatórios criados no OBIEE, e criar relatórios e consultas *Ad-hoc* utilizando o repositório do OBIEE. Também possui um conjunto de funções VBA para o desenvolvimento de macros e aplicações customizadas no Office. O *download* do Smart View pode ser feito através do portal web do OBIEE.

### **BI Baseado em Pesquisa**

O Pentaho não possui um BI baseado em busca para descoberta de dados.

O OBIEE possui busca apenas a objetos criados no OBIEE como relatórios, análises e *dashboard*. Não há uma implementação completa desta funcionalidade para permitir a descoberta de conhecimento através de buscas.

### **BI Móvel**

Para a visualização de conteúdo do OBIEE através de dispositivos móveis é necessário comprar um produto separadamente chamado Oracle BI Mobile. Da mesma forma no Pentaho, é preciso adquirir o produto Pentaho Mobile.

### **3.3.3 Análises**

---

<sup>14</sup> <http://wiki.pentaho.com/display/COM/MS+Project+Integration>

## **OLAP**

O Gartner avalia este aspecto de acordo com a facilidade dos usuários em entender e utilizar as funcionalidades da ferramenta para análises OLAP. É avaliada a performance em diversos atributos, como por exemplo ordenações e ranking sofisticados e performance alternando entre hierarquias.

O Pentaho é altamente capaz de realizar análises OLAP com facilidade e alta performance. As suas ferramentas possibilitam uma excelente experiência ao usuário final. No Pentaho BI Platform existem alguns *plug-ins* (como o Saiku) que permitem uma gama de possibilidades de visões analíticas.

No OBIEE, o Oracle BI Answers permite que sejam usadas operações OLAP como *slicing and dicing*, *drill up* e *drill down*. É possível criar hierarquias e caminhos de *drill* para fornecer a visão multidimensional às análises. Além disso, o OBIEE pode se conectar a bancos multidimensionais, como o Oracle Essbase, para melhorar a performance das consultas. O OBIEE também permite que os usuários gravem dados diretamente nas origens de dados (*write back*).

### **Visualização Interativa**

Para este aspecto, o Gartner procura plataformas que possam fornecer diversos tipos de gráficos além dos básicos e gráficos de “pizza”. Os mapas de calor e geográficos também são levados em conta para esta métrica.

No Pentaho, podemos visualizar alguns tipos de gráfico além dos mais tradicionais. Mas, quando a questão são os mapas de calor e gráficos geolocalizados a solução deixa a desejar.

O OBIEE oferece grande quantidade de formas de visualização dos dados como, por exemplo, gráficos tradicionais, gráficos geográficos, tabelas, tabelas pivot, e manômetros. Estas formas de visualização são interativas possibilitando ao usuário alterar as consultas e salvar uma visão dos dados após a navegação nas análises.

### **Modelagem Preditiva e Mineração de Dados**

O Pentaho é capaz de descobrir conhecimento em bases de dados através da integração com o Weka, que contém ferramentas para pré-processamento de dados, classificação, regressão, clusterização, regras de associação e visualização.

Outra ferramenta é o Pentaho Predictive Analytics, responsável pela modelagem preditiva que é um dos pilares da tomada de decisão.

O OBIEE não possui as funcionalidades de modelagem preditiva e mineração de dados.

### ***Scorecards***

O Pentaho utiliza um *plug-in* chamado STCard, que permite o desenvolvimento de *Scorecards* robustos. Uma das características de STCard, é a flexibilidade para criar diferentes *Scorecards* seja em nível organizacional ou departamental.

O OBIEE utiliza a ferramenta Oracle Scorecard and Strategy Management para a criação de *scorecards*. Esta ferramenta permite a visualização e gestão da estratégia da empresa por área de negócio, área geográfica ou outras estruturas organizacionais. Possibilita alinhar a direção estratégica da empresa e comunicar a todos na organização.

## **3.4 Tabelas Comparativas**

A seguir, são apresentadas três tabelas tipo *checklist*, uma para cada categoria de métricas, que mostram quais itens são contemplados pelas plataformas e quais itens as plataformas não possuem.

<b>Integração</b>		
	<b>OBIEE</b>	<b>Pentaho</b>
<b>Infraestrutura de BI</b>	S	S
<b>Gerenciamento de Metadados</b>	S	S
<b>Ferramentas de Desenvolvimento</b>	S	S
<b>Colaboração</b>	S	N

Tabela 1: *Checklist* da categoria Integração

## **Entrega de Informação**

	<b>OBIEE</b>	<b>Pentaho</b>
<b>Relatórios</b>	S	S
<b>Dashboards</b>	S	S
<b>Consultas Ad hoc</b>	S	S
<b>Integração com Microsoft Office</b>	S	N
<b>BI Baseado em Pesquisa</b>	N	N
<b>BI Móvel</b>	S	S

Tabela 2: *Checklist* da categoria Entrega de Informação

<b>Análise</b>		
	<b>OBIEE</b>	<b>Pentaho</b>
<b>OLAP</b>	S	S
<b>Visualização Interativa</b>	S	S
<b>Modelagem Preditiva e Mineração de Dados</b>	N	S
<b>Scorecards</b>	S	S

Tabela 3: *Checklist* da categoria Análise

## 4 Estudo de Caso: A Base de dados da Copa 2014

### 4.1 Introdução

Embora o foco deste projeto não seja uma análise completa de uma base de dados a fim de tomar decisões, escolhemos um assunto recorrente nos últimos anos e que desperta o interesse da população brasileira, pelo aspecto de transparência que a sociedade requer: a Copa do Mundo de Futebol de 2014.

O Brasil tem investido cada vez mais em políticas de acesso à informação. De acordo com a Lei n 12.527, Lei de Acesso à Informação (LAI), "é dever dos órgãos e entidades públicas promover, independentemente de requerimentos, a divulgação em local de fácil acesso, no âmbito de suas competências, de informações de interesse coletivo ou geral por eles produzidas ou custodiadas".

A divulgação das informações deve ser feita através de sítios oficiais na Internet e os dados devem estar disponíveis obedecendo aos seguintes requisitos.

- Possibilitar a gravação de relatórios em diversos formatos eletrônicos, inclusive abertos e não proprietários, tais como planilhas e texto, de modo a facilitar a análise das informações.
- Possibilitar o acesso automatizado por sistemas externos em formatos abertos, estruturados e legíveis por máquina.
- Divulgar em detalhes os formatos utilizados para estruturação da informação.<sup>15</sup>

A disponibilidade e o fácil acesso aos dados foram alguns dos critérios para a escolha da base. Embora haja uma familiaridade com o assunto Copa do Mundo, as

---

<sup>15</sup> Pedroso, L., Tanaka, A. and Cappelli, C. (2013) "A Lei de Acesso à Informação Brasileira e os Desafios Tecnológicos dos Dados Abertos Governamentais", <http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2013/0048.pdf>, May.

bases tratam de questões financeiras mais técnicas referentes a orçamentos e valores de empreendimentos.

O site Copa 2014 - Transparência <sup>16</sup> em 1º lugar é uma iniciativa do Governo Federal que tem o objetivo de dar transparência às ações e despesas do Poder Executivo Federal relativas aos jogos da Copa do Mundo de Futebol de 2014, seja na forma de gastos diretos ou de transferências aos estados e municípios envolvidos com o evento.

Por ser uma base ampla, é possível derivar diversas análises interessantes. Mas, como já foi mencionado anteriormente, neste projeto queremos apenas mostrar a comparação entre duas soluções de BI. Sendo assim, definimos apenas um nicho que queremos visualizar e não optamos por análises complexas.

A manutenção do site está sob a responsabilidade da Controladoria-Geral da União (CGU), <sup>17</sup> órgão que trabalha com a defesa do patrimônio público e com o incremento da transparência da gestão, por meio das atividades de controle interno, auditoria pública, correição, prevenção e combate à corrupção, e ouvidoria.

## **4.2 Processo de ETL**

Os dados da Copa estão disponibilizados em sua forma primária bruta para permitir o maior número possível de levantamentos e análises. Sendo assim, as agregações e derivações que possam surgir a partir destes ficam a cargo do utilizador.

Os dados estão organizados em 50 arquivos do tipo TXT e 50 arquivos do tipo CSV. Cada tabela do banco de dados possui um arquivo TXT e um arquivo CSV. Os arquivos TXT descrevem as tabelas. Nesses arquivos, estão contidos o nome, tipo de dado e uma breve descrição de cada coluna de uma tabela. Os arquivos CSV contêm os dados das tabelas. Para este projeto, foram utilizados dados atualizados até outubro de 2013.

Não foi encontrada uma descrição da base de dados com mais informações sobre a sua estrutura, como quais são as chaves estrangeiras de cada tabela ou quais são os relacionamentos existentes na base. Por isso, foi preciso analisar os arquivos TXT para descobrir os relacionamentos e chaves estrangeiras das tabelas.

---

<sup>16</sup> <http://www.portaltransparencia.gov.br/copa2014/home.seam>

<sup>17</sup> <http://www.portaltransparencia.gov.br/copa2014/saibamais.seam?textoIdTexto=23>

O padrão usado para construir os relacionamentos entre as tabelas foi verificar se uma tabela possuía em seu arquivo TXT uma coluna com nome igual a uma chave primária de outra tabela.

Para a carga de dados, primeiramente foi criada uma *batch* no Windows para juntar todos os arquivos CSV de uma tabela em apenas um arquivo. Em seguida, foi utilizado o Kettle para carregar os dados dos arquivos criados pela *batch* no banco de dados. O Kettle é a ferramenta de ETL da plataforma de BI Pentaho. Essa ferramenta foi de extrema importância para que pudéssemos carregar corretamente os dados no banco.

### **4.3 Modelagem Dimensional – Esquema Estrela**

Com o banco de dados criado e os dados carregados foi possível iniciar o estudo do banco de dados e de sua estrutura para definir o esquema estrela que seria usado no desenvolvimento do projeto.

Após um primeiro estudo, foi definido que seriam criados dois esquemas estrela, um para armazenar os dados de orçamento e outro para armazenar os dados do realizado. Porém essa ideia foi abandonada, pois verificamos que tornaria o projeto muito complexo, fugindo ao foco estabelecido que está na comparação de duas plataformas de BI e não em uma análise profunda dos dados da Copa 2014. Assim, decidimos criar apenas um esquema estrela com dados do realizado, como prova de conceito do desenvolvimento do projeto.

O esquema estrela criado possui quatro tabelas de dimensão e uma tabela fato. As dimensões são: Instituição, Tempo, Andamento Financeiro, e Empreendimento. A tabela fato é: Execução Financeira.

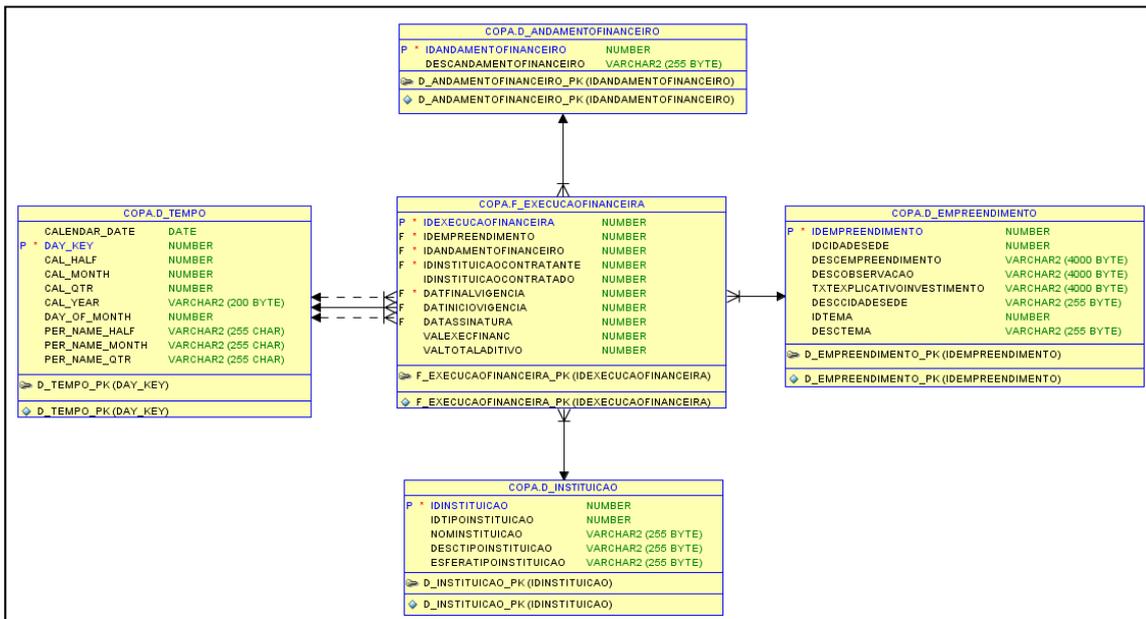


Figura 19: Esquema estrela usado no projeto

#### 4.3.1 Dimensão Empreendimento

A dimensão Empreendimento possui colunas que descrevem um empreendimento da Copa 2014 como, por exemplo, a reforma do Estádio Maracanã ou Reforma do Terminal de Passageiros 2 do Aeroporto Internacional do Rio de Janeiro/Galeão.

Coluna	Descrição
IDEMPREENDIMENTO	Identificador do empreendimento
IDCIDADESEDE	Identificador da cidade sede
DESCEMPREENDIMENTO	Descrição (nome) do empreendimento
DESCOBSERVACAO	Observações sobre o Empreendimento
TXTEXPLICATIVOINVESTIMENTO	Texto explicativo sobre os Investimentos do Empreendimento
DESCCIDADESEDE	Descrição (nome) da cidade sede
IDTEMA	Identificador do tema
DESCTEMA	Descrição (nome) do tema. Exemplos: Aeroportos, Estádios.

Tabela 4: Descrição da Dimensão Empreendimento

#### 4.3.2 Dimensão Instituição

A dimensão Instituição possui colunas para descrever as instituições envolvidas nos empreendimentos da Copa 2014, sejam elas organizações contratantes como o Governo do Estado do Rio de Janeiro, ou empresas contratadas.

<b>Coluna</b>	<b>Descrição</b>
IDINSTITUICAO	Identificador da instituição
IDTIPOINSTITUICAO	Identificador do tipo de instituição
NOMINSTITUICAO	Nome da instituição.
DESCTIPOINSTITUICAO	Descrição (nome) do tipo de instituição. Exemplos: Governo Estadual, Entidade Privada.
ESFERATIPOINSTITUICAO	Descrição (nome) da esfera da instituição: Exemplos: Estado, Setor privado.

Tabela 5: Descrição da Dimensão Instituição

### 4.3.3 Dimensão Tempo

A dimensão Tempo possui colunas que descrevem a data de assinatura do contrato de um determinado empreendimento. O usuário final pode filtrar as consultas por ano, semestre, trimestre, mês e dia.

<b>Coluna</b>	<b>Descrição</b>
CALENDAR_DATE	Data no formato DD/MM/AAAA
DAY_KEY	Número identificador da data no formato AAAAMMDD.
CAL_HALF	Número do semestre
CAL_MONTH	Número do mês
CAL_QTR	Número do trimestre
CAL_YEAR	Número do ano
DAY_OF_MONTH	Número do dia
PER_NAME_HALF	Semestre no formato "Ano" mais "Número do Semestre" mais a palavra "Semestre"

PER_NAME_MONTH	Nome do mês
PER_NAME_QTR	Trimestre no formato "Ano" mais a letra "T" mais "número do trimestre"

Tabela 6: Descrição da Dimensão Tempo

#### 4.3.4 Dimensão Andamento Financeiro

A dimensão Andamento Financeiro possui colunas que descrevem o status de uma execução financeira (Não Iniciado, Em andamento, Concluído, Suspenso).

Coluna	Descrição
IDANDAMENTOFINANCEIRO	Identificador do andamento financeiro
DESCANDAMENTOFINANCEIRO	Texto que descreve o andamento financeiro

Tabela 7: Descrição da Dimensão Andamento Financeiro

#### 4.3.5 Fato Execução Financeira

A tabela fato Execução Financeira possui medidas sobre as execuções financeiras de um determinado empreendimento.

Coluna	Descrição
IDEXECUCAOFINANCEIRA	Chave primária de Execução Financeira
IDEMPREENDIMENTO	Chave estrangeira para a tabela de dimensão Empreendimento
IDANDAMENTOFINANCEIRO	Chave estrangeira para a tabela de dimensão Andamento Financeiro
IDINSTITUICAOCONTRATANTE	Chave estrangeira para a tabela de dimensão Instituicao. Identifica qual é a instituição contratante
IDINSTITUICAOCONTRATADO	Chave estrangeira para a tabela de dimensão Instituição. Identifica qual é a instituição contratada
DATFINALVIGENCIA	Chave estrangeira para a tabela de dimensão Tempo. Identifica a data final de vigência de uma execução financeira

DATINICIOVIGENCIA	Chave estrangeira para a tabela de dimensão Tempo. Identifica a data de início de vigência de uma execução financeira
DATASSINATURA	Chave estrangeira para a tabela de dimensão Tempo. Identifica a data de assinatura do contrato de uma execução financeira
VALEXECFINANC	Valor da execução financeira
VALTOTALADITIVO	Valor agregado dos aditivos da execução financeira

Tabela 8: Descrição da tabela Fato Execução Financeira

## 5 Desenvolvimento no OBIEE

### 5.1 Criando o repositório

#### 5.1.1 Camada física

Para a criação da camada física, primeiramente deve ser criada uma conexão com o banco de dados. O OBIEE fornece diversas formas de conexão às fontes de dados. A utilizada neste desenvolvimento será a Oracle Call Interface (OCI), pois o banco de dados utilizado é o Oracle Database 11g e essa é a forma recomendada pela Oracle para conexão aos seus banco de dados.

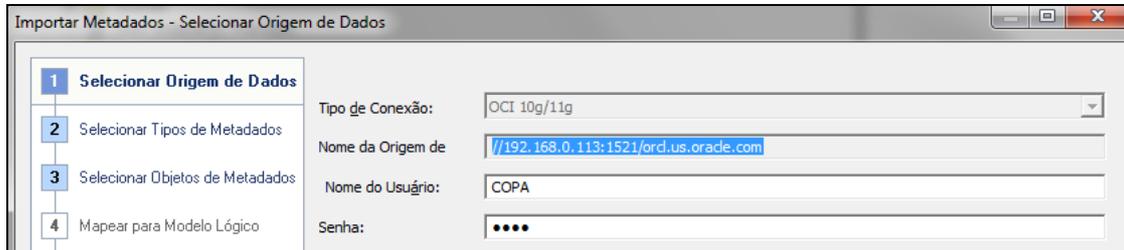


Figura 20: Campos para conexão utilizando OCI

Depois de estabelecida a conexão, são escolhidos os objetos que irão compor a camada física do repositório. O repositório criado neste projeto irá utilizar apenas tabelas do banco de dados relacional. As tabelas selecionadas foram: "D\_ANDAMENTOFINANCEIRO", "D\_EMPREENDIMENTO", "D\_INSTITUICAO", "D\_TEMPO" e "F\_EXECUCAOFINANCEIRA". Depois de selecionar as tabelas, o processo de importação dos metadados para o repositório do OBIEE é iniciado.

Após a importação, a primeira tarefa a fazer é criar tabelas *Alias* para cada uma das tabelas importadas. As tabelas *alias* possuem a mesma estrutura das tabelas físicas. São usadas para a eliminação de relacionamentos desnecessários existentes entre as tabelas físicas e para a utilização de nomes padronizados.

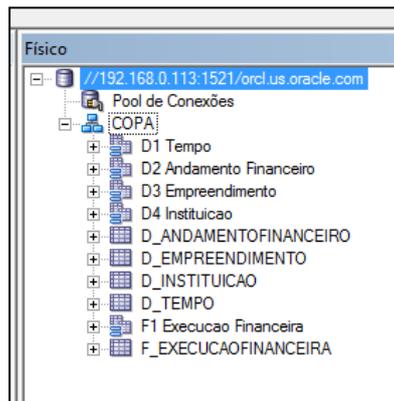


Figura 21: Tabelas Físicas e Tabelas *Alias* na camada física

A seguir, são definidos os relacionamentos entre as tabelas *Alias*. Isso é feito graficamente, sem precisar digitar comandos SQL. A figura abaixo mostra o esquema estrela com as tabelas e seus relacionamentos definidos.

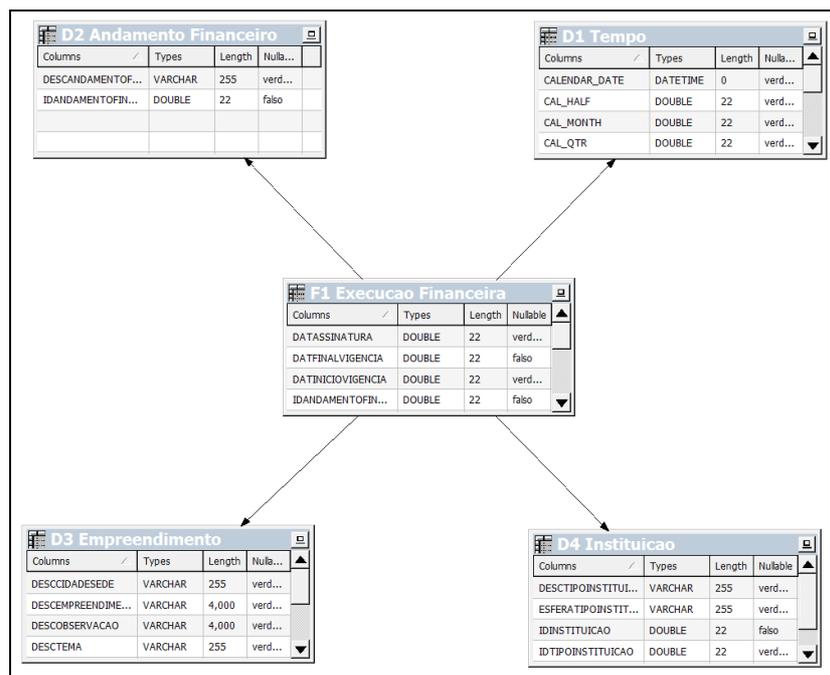


Figura 22: Esquema estrela inicial do repositório

### 5.1.2 Camada de modelo de negócios e mapeamento

Com a camada física criada, é possível criar a estrutura da camada de modelo de negócios e mapeamento. Primeiramente, é criado o modelo de negócios e, em seguida, são inseridas as tabelas *Alias* da camada física. Para fazer isso, basta selecionar as tabelas *Alias* da camada física e arrastá-las para dentro do modelo. Se todas as tabelas

forem inseridas ao mesmo tempo, seus relacionamentos na camada de modelo de negócio e mapeamento são criados automaticamente.

Para melhor visualização dos campos das tabelas, foram realizadas renomeações nesses campos. Para isso, foi inserido o caractere "\_" entre as palavras de um determinado campo. Por exemplo: "DESCCIDADESEDE" foi renomeado para "DESC\_CIDADE\_SEDE". Em seguida, foi utilizado o assistente de renomeação do Administration Tool. No assistente, foram selecionadas as opções para colocar todas as letras em minúsculo, substituir caracteres "\_" por espaço e colocar letra maiúscula no início de cada palavra. O campo "DESC\_CIDADE\_SEDE", por exemplo, foi renomeado para "Desc Cidade Sede".

Feito isso, foi configurada a forma de agregação de cada medida da tabela Fato na camada de modelo de negócio e mapeamento. A tabela fato possui duas medidas: "Val Exec Financ" e "Val Total Aditivo". Selecionamos a soma como forma de agregação para as duas medidas.

A figura abaixo mostra o resultado da modelagem da camada de modelo de negócios e mapeamento.

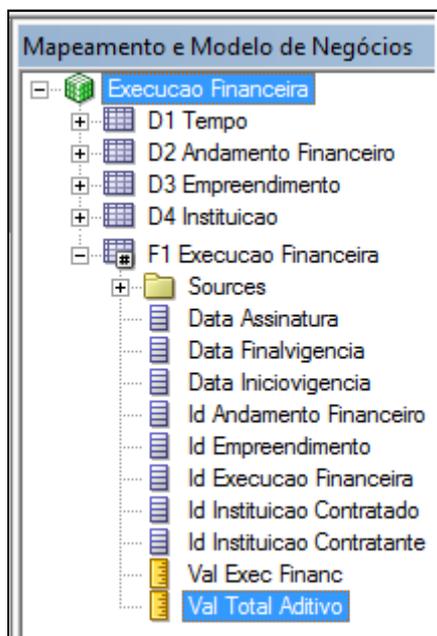


Figura 23: Camada de Modelo de Negócios e Mapeamento

### 5.1.3 Camada de apresentação

Com o modelo de negócios pronto, é criada a área de assunto na camada de apresentação e são inseridas as tabelas da camada de modelo de negócios e mapeamento. Após a inserção das tabelas, são removidos os campos de ID e outros campos desnecessários. Também são feitas renomeações em alguns campos e reordenação dos campos nas tabelas.

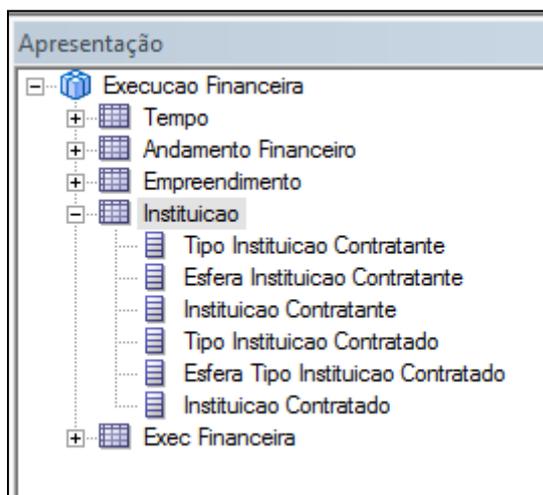


Figura 24: Camada de Apresentação

#### 5.1.4 Criando medidas calculadas

É possível criar novas medidas na camada de modelo de negócios e mapeamento utilizando medidas existentes na tabela fato. Para ilustrar essa funcionalidade, foi criada a medida "Val Exec Financ mais Aditivos". Esta medida é o resultado da soma das medidas "Val Exec Financ" e "Val Total Aditivo". Com isso, é possível consultar o valor total gasto em uma execução financeira incluindo os aditivos.

Para criar a nova medida, é inserida uma fórmula na medida. Esta fórmula é a soma das duas medidas existentes na tabela fato e pode ser criada de maneira gráfica ou não.

Após criar a nova medida no modelo de negócios, basta arrastá-la para a camada de apresentação para torná-la disponível ao usuário final.

#### 5.1.5 Criando Hierarquias

Para fornecer uma visão OLAP do modelo de negócios, permitindo operações de *drill*, por exemplo, são definidas hierarquias para cada tabela de dimensão. As hierarquias são definidas na camada de modelo de negócios e mapeamento.

Primeiramente, utilizando uma tabela de dimensão do modelo de negócios como origem, é criada uma dimensão lógica. A seguir, são criados os níveis dessa dimensão e, para cada nível, são associados um ou mais campos provenientes da tabela de dimensão de origem.

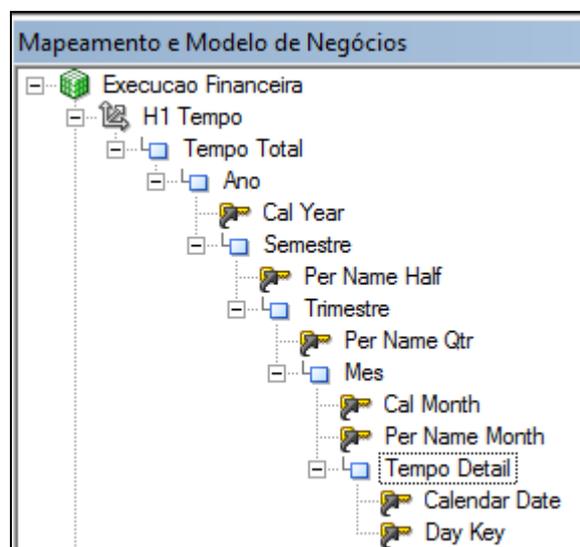


Figura 25: Dimensão Lógica de Tempo

Nesta parte do desenvolvimento foi necessário modificar o esquema estrela da camada de negócios do OBIEE pelo seguinte motivo: a tabela fato possui dois campos, "Id Instituicao Contratado" e "Id Instituicao Contratante", relacionados com o campo "Id Instituicao" da tabela "Instituicao", caso típico de dimensão exercendo dois papéis.

Para modelar este tipo dimensão no OBIEE, foi feita a seguinte alteração: a tabela *alias* "D4 Instituicao" na camada física foi substituída por duas tabelas *alias*: "D40 Instituicao Contratante" e "D41 Instituicao Contratado". Essas tabelas *alias* tem como origem a tabela física "Instituicao", ou seja, não foi necessário criar duas tabelas no banco de dados relacional.

Na camada de modelo de negócios, a tabela "Instituicao" foi removida e foram inseridas as duas novas tabelas da camada física: "Instituicao Contratante" e "Instituicao Contratada". Após essa nova modelagem, foi possível criar dimensões lógicas para essas tabelas e terminar a modelagem de hierarquias na camada de modelo de negócios.

A figura abaixo mostra como ficou o novo esquema estrela na camada de modelo de negócios e mapeamento. O esquema estrela no nível do banco de dados relacional não foi modificado.

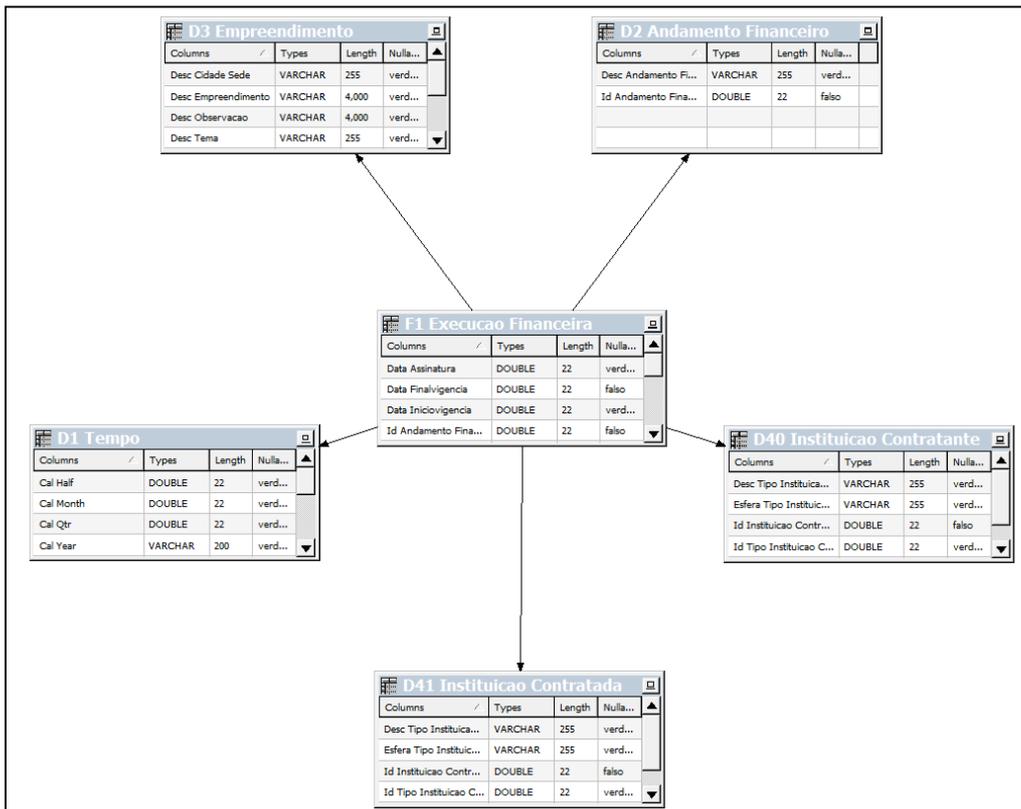


Figura 26: Camada de Modelo de Negócios e Mapeamento - Esquema Estrela

Após a criação de uma dimensão lógica para cada tabela de dimensão do modelo de negócios, as dimensões lógicas são inseridas na camada de apresentação.

### 5.1.6 Utilizando tabelas com dados pré-agregados

Nesta etapa do desenvolvimento são criadas tabelas com dados pré-agregados. Essas tabelas têm como objetivo tornar mais rápidas as consultas em dados agregados como, por exemplo, consultas por ano ou trimestre.

Primeiramente, foram criadas duas tabelas no banco de dados relacional: "D\_TEMPO\_QTR\_GRAIN" e "F\_EXECUCAOFINANCEIRA\_AGG". A tabela "D\_TEMPO\_QTR\_GRAIN" é quase igual a tabela de dimensão "D\_TEMPO" utilizada previamente, a diferença é que sua granularidade é a nível de trimestre.

A tabela "F\_EXECUCAOFINANCEIRA\_AGG" possui os dados das duas medidas do esquema estrela agregados por trimestre. Também possui uma coluna com os IDs dos empreendimentos e uma coluna com os IDs dos trimestres.

Após a criação das tabelas, é feita a importação dessas duas tabelas para a camada física do repositório do OBIEE. Em seguida são criadas as tabelas *alias* para cada tabela

e é feito o relacionamento entre três tabelas. A tabela *alias* "F2 Execucao Financeira Agregada" possui relacionamento com as tabelas *alias* "D1 Tempo Trimestre Agregado" e "D3 Empreendimento".

Em seguida, cada campo da tabela *alias* "D1 Tempo Trimestre Agregado" é arrastado sobre o campo correspondente da tabela "D1 Tempo" da camada de modelo de negócios. Isso fará com que um campo na tabela "D1 Tempo" da camada de modelo de negócios esteja associado à duas tabelas físicas diferentes. O mesmo é feito entre a tabela *alias* "F2 Execucao Financeira Agregada" e a tabela "F1 Execucao Financeira" do modelo de negócios.

Quando o usuário fizer uma consulta por trimestre, empreendimento e valor de aditivos, por exemplo, o BI Server irá realizar a consulta utilizando as tabelas agregadas. As tabelas agregadas são utilizadas nas consultas por trimestre, semestre ou ano, contendo ou não campos da tabela de empreendimento. Caso o usuário faça uma consulta por mês, o BI Server irá consultar as tabelas físicas não agregadas.

### 5.1.7 Utilizando variáveis

Variáveis podem ser usadas para facilitar tarefas administrativas e mudar dinamicamente o conteúdo de metadados. Podem ser usadas em expressões e fórmulas criadas no Administration Tool ou em análises, relatórios e *dashboards*.

Para exemplificar essa funcionalidade, foram criadas duas variáveis. Uma variável contém o ano atual. A outra variável contém o trimestre atual.

As variáveis podem ser criadas utilizando comandos SQL para retornar o código do trimestre atual e o ano atual da tabela de tempo do esquema estrela. O comando SQL usado foi o seguinte:

```
SELECT PER_NAME_QTR, CAL_YEAR  
FROM COPA.D_TEMPO  
WHERE  
TO_CHAR(CALENDAR_DATE, 'DD/MM/YYYY')=(SELECT  
TO_CHAR(SYSDATE, 'DD/MM/YYYY') FROM dual);
```

A variável de trimestre irá receber o valor da primeira coluna da consulta e a variável de ano irá receber o valor da segunda coluna.

### 5.1.8 Criando medidas de séries temporais

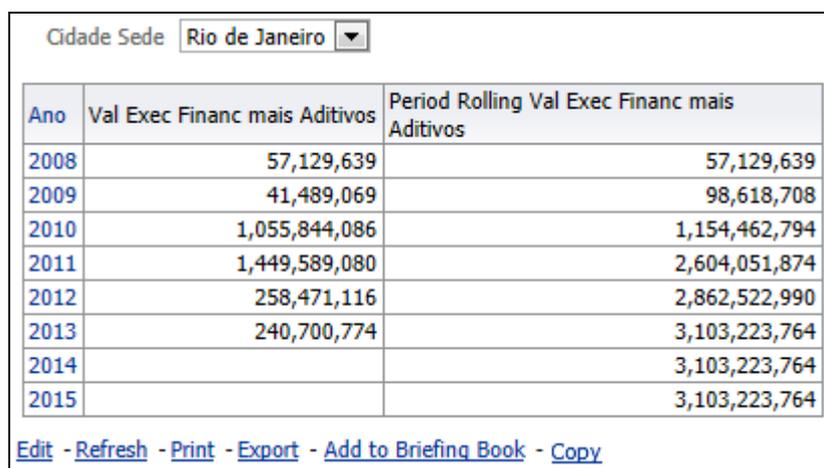
Uma série de tempo é uma sequência de dados, medidos em pontos sucessivos no tempo, espaçados em intervalos de tempo. O OBIEE possui funções que permitem a criação de medidas de séries temporais.

Para exemplificar essa funcionalidade, foi criada uma nova medida que utiliza a função Period Rolling e a medida "Val Exec Financ mais Aditivos". Essa função soma um determinado intervalo de tempo.

Primeiramente, é criada uma nova medida na tabela fato com o nome "Period Rolling Val Exec Financ mais Aditivos" e é criada uma expressão que usa a função Period Rolling para somar os valores dos cem últimos períodos. Em seguida, na dimensão lógica de tempo, são definidas chaves cronológicas para cada nível da dimensão. Essas chaves servirão para definir a ordem dos valores em cada nível.

Por fim, a nova medida é inserida na camada de apresentação.

A figura abaixo ilustra a utilização da nova medida. A segunda coluna possui a medida de série temporal. Pode-se observar que os valores vão se acumulando a cada ano. Por exemplo, o valor do ano de 2013 na segunda coluna é igual a soma dos valores dos anos de 2008, 2009, 2010, 2011, 2012 e 2013 da primeira coluna.



Ano	Val Exec Financ mais Aditivos	Period Rolling Val Exec Financ mais Aditivos
2008	57,129,639	57,129,639
2009	41,489,069	98,618,708
2010	1,055,844,086	1,154,462,794
2011	1,449,589,080	2,604,051,874
2012	258,471,116	2,862,522,990
2013	240,700,774	3,103,223,764
2014		3,103,223,764
2015		3,103,223,764

[Edit](#) - [Refresh](#) - [Print](#) - [Export](#) - [Add to Briefing Book](#) - [Copy](#)

Figura 27: Exemplo de medida de séries temporais

## 5.2 Análises, Dashboards e Relatórios

O ambiente de criação de análises e dashboards no OBIEE é acessado através do navegador web. O usuário realiza login em um portal e tem acesso às funcionalidades

oferecidas pela suíte para visualização dos dados. A figura abaixo mostra a interface do portal web do OBIEE.

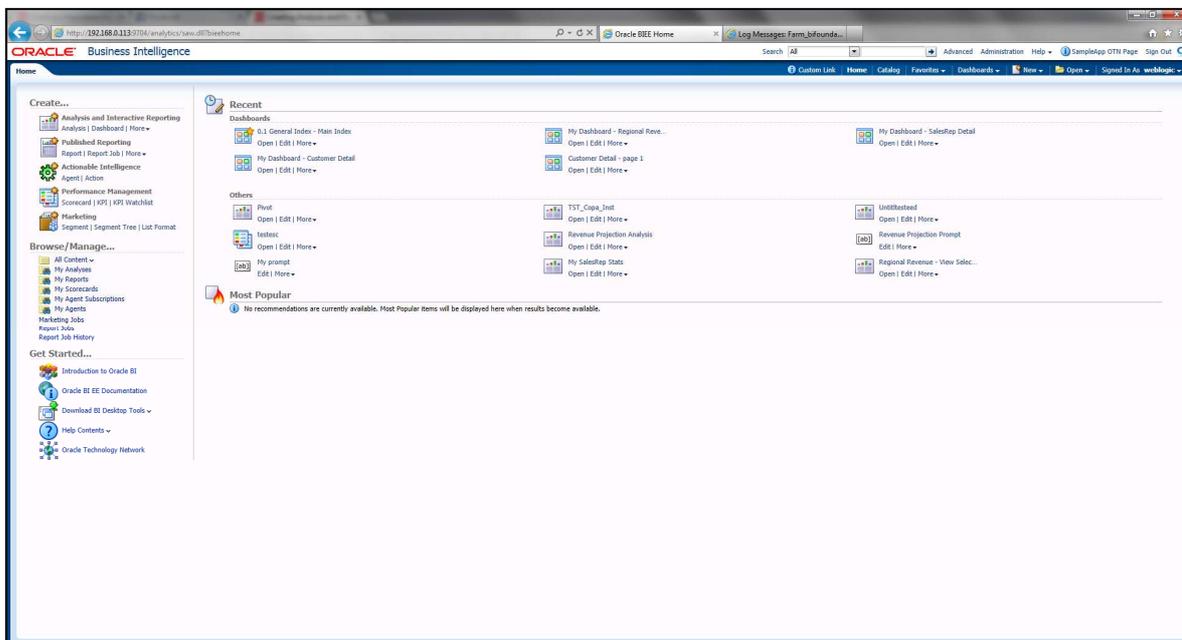


Figura 28: Portal do OBIEE

Nesta página web é possível criar análises, *dashboards*, relatórios, *scorecards*, realizar *download* de *clients desktop*, abrir relatórios visualizados recentemente e realizar buscas.

Os usuários podem adicionar análises e *dashboards* aos favoritos, permitindo acesso rápido através do link "Favorites" no topo da página inicial do OBIEE.

### 5.2.1 Criando Análises

A área de criação de análises possui quatro abas: Criteria, Results, Prompts, e Advanced.

Na aba Criteria são definidos quais dimensões e campos a consulta possuirá. Também é possível definir filtros para restringir os metadados e dados da consulta, definir a forma de ordenação dos dados, e configurar as propriedades das colunas. O usuário também pode criar colunas com fórmulas calculadas no relatório, ou seja, é possível criar novas medidas dentro de um relatório utilizando medidas existentes e fórmulas.

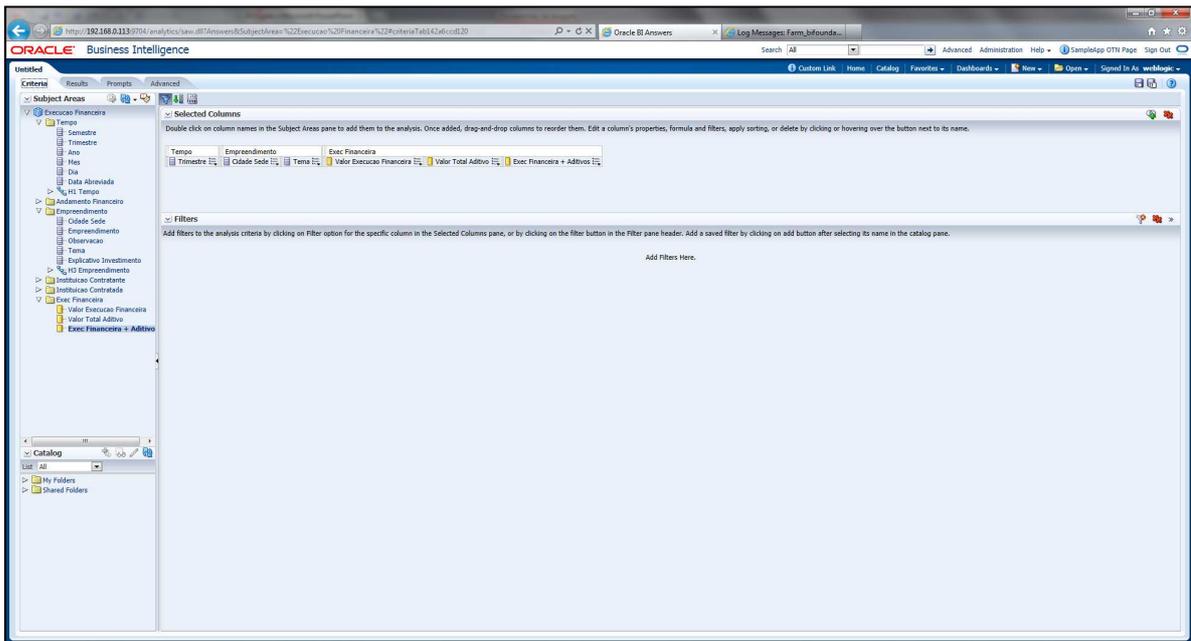


Figura 29: Aba *Criteria* da área de criação de análises

Na aba *Results* o usuário pode visualizar o resultado da consulta, inserir e excluir campos, editar propriedades da análise, imprimir a análise e inserir e editar títulos, tabelas, tabelas *pivot*, gráficos, manômetros, narrativas, e outras formas de visualização. Nesta aba também é possível exportar a análise para PDF, Excel, Powerpoint, Arquivo Web (.MHT), arquivo .CSV, arquivo delimitado por tabulações, e arquivo XML.

O usuário pode relacionar dois tipos de visualização de dados para que quando os metadados de uma visualização forem alterados (através de *drill*, por exemplo) o mesmo ocorrerá com a visualização dependente. Por exemplo, pode-se criar uma tabela e um gráfico. Quando for feito um *drill down* no campo "cidade sede" da tabela para visualizar empreendimentos, o gráfico também será alterado automaticamente para mostrar os dados por empreendimento.

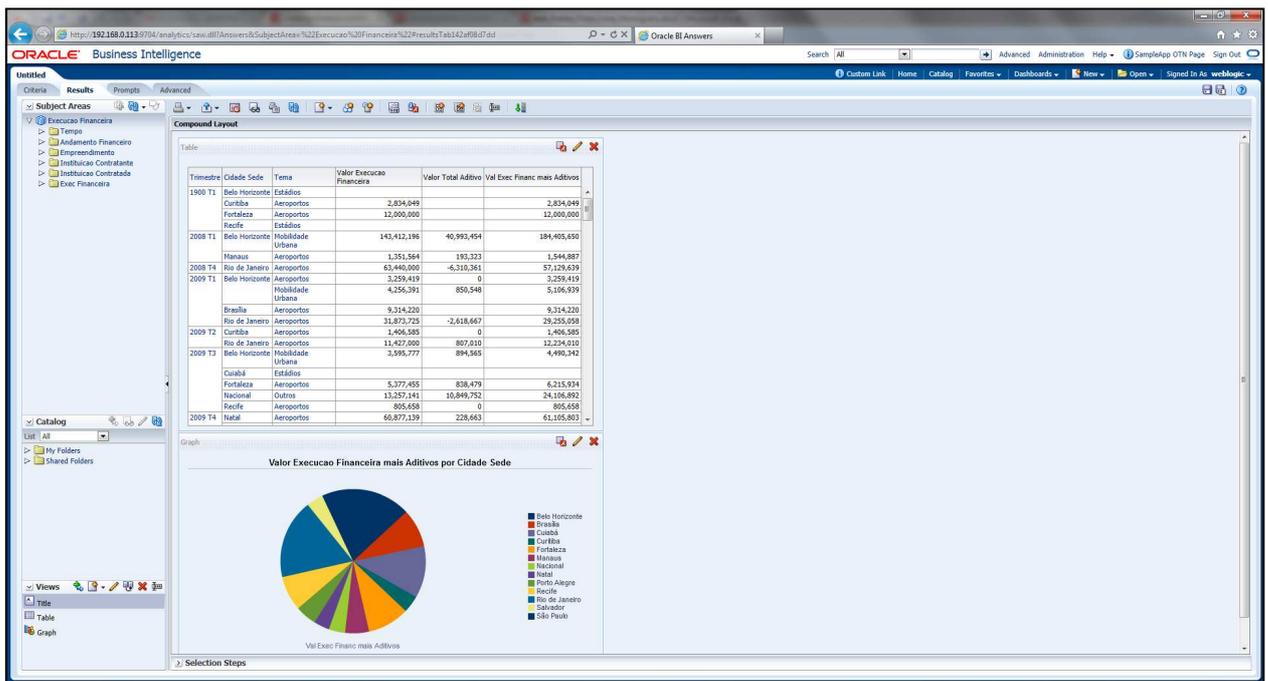


Figura 30: Aba Results da área de criação de análises

Na aba *Prompts*, é possível criar *prompts* para definir quais dados irão fazer parte da visualização. Quando o usuário abrir uma análise, será pedido que preencha quais campos deseja visualizar nas dimensões.

Na aba *Advanced*, usuários mais experientes e com os privilégios adequados podem visualizar e editar o código XML e a consulta SQL lógica gerada pela a análise.

A figura abaixo apresenta o resultado final de uma análise criada no OBIEE.

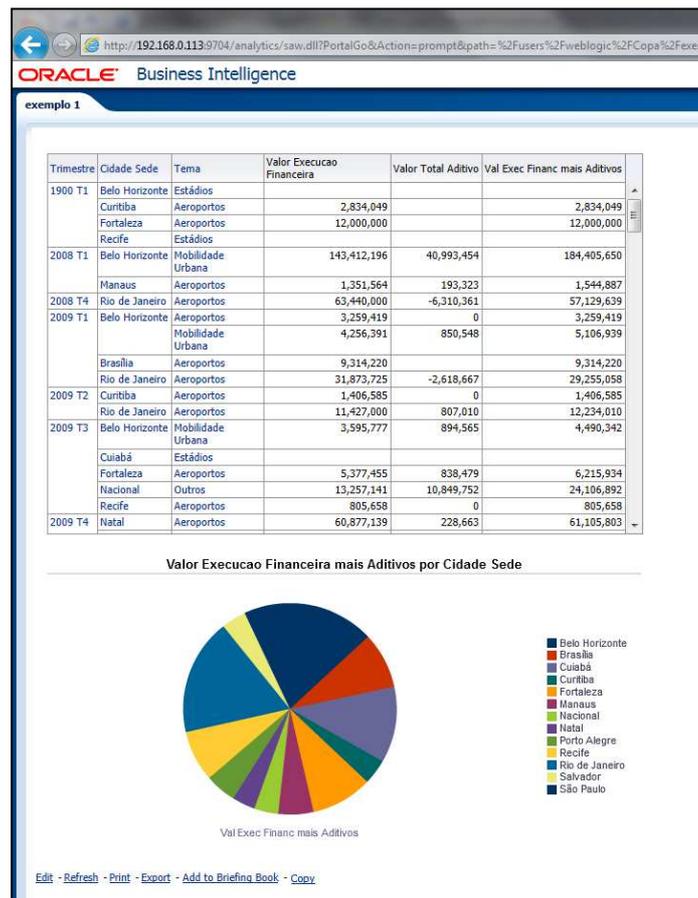


Figura 31: Exemplo de Análise Criada no OBIEE

## 5.2.2 Criando Dashboards

A área de edição de *dashboards* possui os objetos disponíveis para inserção no *dashboard* como colunas, seções, alertas, ações, links, imagens, textos e pastas. Nesta página também é possível navegar no catálogo de objetos para inserir rapidamente análises, filtros, *prompts* e outros objetos criados previamente. Um *dashboard* pode possuir diversas páginas que serão apresentadas como abas na visualização.

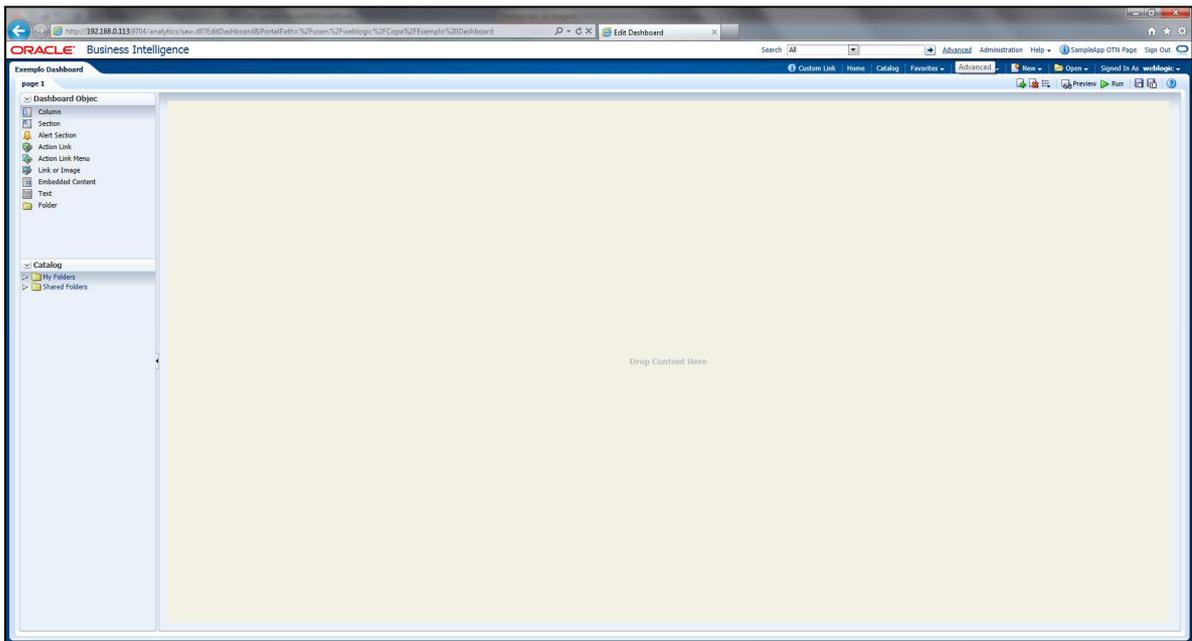


Figura 32: Área de Edição de *Dashboards*

A organização do *dashboard* é feita arrastando os objetos para dentro do *dashboard*. Um *dashboard* é organizado inserindo colunas e seções dentro das colunas. Dentro das seções podem ser inseridos textos, alertas, links, imagens, pastas e as análises previamente criadas. É possível inserir também filtros, *prompts* e variáveis para modificar dinamicamente os dados de uma consulta. Cada coluna e seção possuem suas próprias configurações.

É possível inserir condições em cada seção para definir se esta deverá ser apresentada no *dashboard* ou não. As condições também podem determinar se *links* de ação deverão ser mostrados, e se alertas devem ser enviados para usuários finais.

O usuário final pode escolher um *dashboard* para aparecer na tela inicial do OBIEE assim que efetuar *login*. Também pode salvar suas preferências para um determinado *dashboard* e salvar uma visão do *dashboard* após realizar uma navegação, de modo que possa escolher uma visão salva e acessá-la rapidamente.

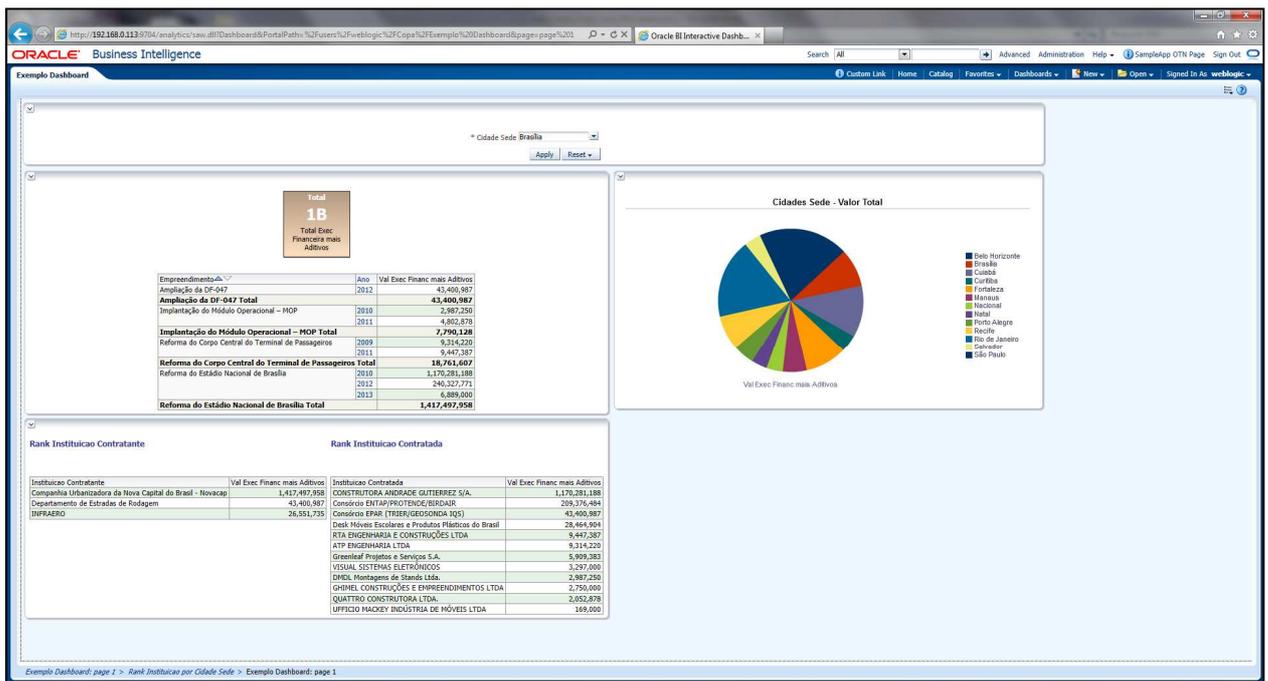


Figura 33: Exemplo de *Dashboard* criado no OBIEE

### 5.2.3 Criando Relatórios

Para criar um relatório no OBIEE, primeiramente é escolhida a origem dos dados. As opções de origem de dados são: data model, planilha excel, e área de assunto da camada de apresentação.

Após a definir a origem dos dados, o usuário pode escolher entre dois métodos de criação de relatórios: guiado ou utilizando o editor de relatórios.

No método guiado, o relatório é criado em, no máximo, seis passos. O número de passos varia de acordo com o layout escolhido. O primeiro passo é a escolha da origem de dados. Este passo é comum aos dois métodos de criação. No segundo passo são escolhidas opções de página e é definido o layout do relatório. Existem alguns layouts prontos, mas também é possível criar layouts do tipo RTF, Excel, PDF, Flash, e eText.

No terceiro passo, é escolhido o tipo de gráfico (colunas, linhas ou pizza) e são definidos os campos que irão compor o gráfico. No quarto passo, são definidas as configurações para o segundo gráfico, caso o layout escolhido possua dois gráficos.

No quinto passo, são definidos os campos que irão compor a tabela. Por fim, no sexto passo, o usuário pode escolher em visualizar o relatório ou continuar a edição utilizando o editor de relatórios.

A figura abaixo mostra um relatório criado no modo guiado e que tem como origem de dados a camada de apresentação criada anteriormente neste trabalho.

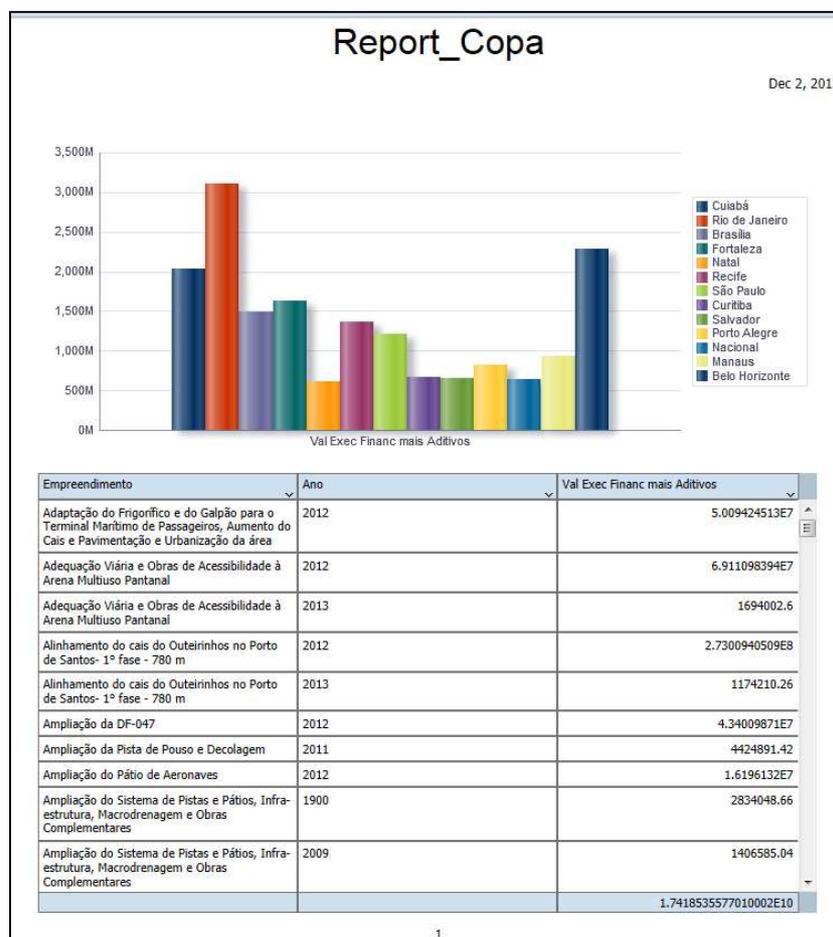


Figura 34: Relatório criado no modo guiado

É possível utilizar o editor de relatórios para alterar formatações, e adicionar e remover campos e objetos do relatório.

A figura a seguir mostra o relatório da figura anterior editado no editor de relatórios. Foram feitas as seguintes edições:

- Alterado o texto do título
- Adicionado título para o gráfico
- Inserido filtro para o usuário escolher uma cidade sede na tabela
- Alterado o tamanho das colunas da tabela
- Alterado tipo de dados da terceira coluna da tabela para número

- Formateado em negrito a linha de total da tabela e inserido o texto "Total" nesta linha
- Destacados valores maiores que 200.000.000
- Agrupados os nomes dos empreendimentos na tabela

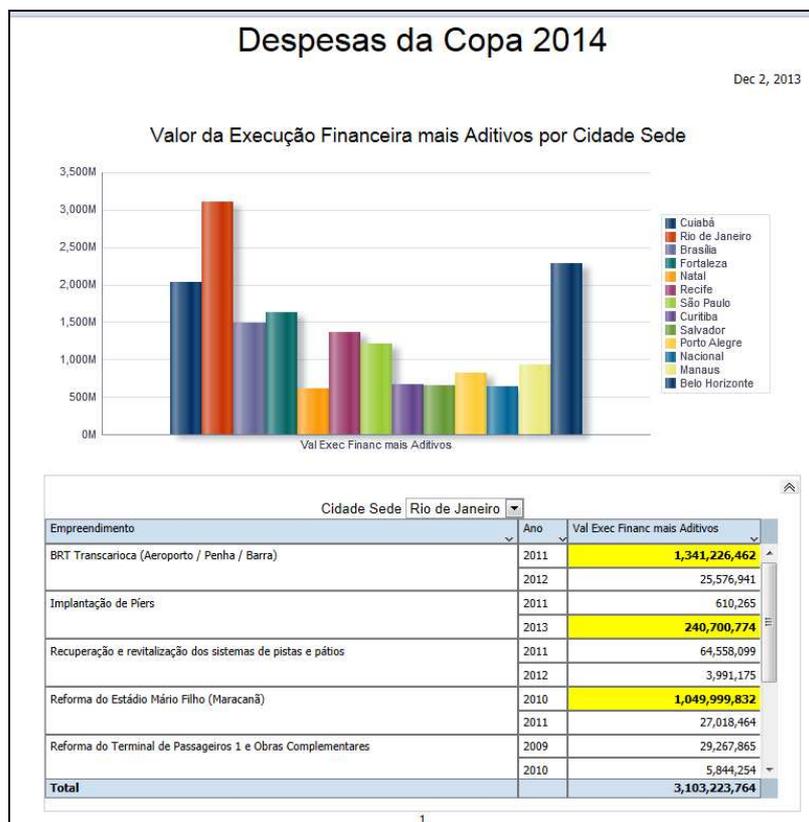


Figura 35: Relatório editado no editor de relatórios

Após a criação de um relatório, é possível agendá-lo para distribuição para usuários finais. Os relatórios podem ser enviados para os seguintes destinos: impressão, pasta na Web, e-mail, fax, e FTP. Os relatórios podem ser enviados nos formatos PDF, HTML, RTF, Excel e PowerPoint.

### 5.3 Alertas e Agentes

O OBIEE possui a funcionalidade de enviar alertas aos usuários finais. Para enviar alertas é necessário criar um agente. Agentes possibilitam a automatização de processos de negócio. Podem enviar alertas e objetos do catálogo do OBIEE como, por exemplo, análises e dashboards, aos usuários finais. Os objetos podem ser enviados em diversos formatos como HTML, PDF, MS Excel e MS PowerPoint. Os destinos dos alertas

podem ser a página inicial do OBIEE e dashboard, e-mail, Pager, telefones digitais e dispositivos móveis.

Um agente também pode executar uma determinada ação como invocar um *web service* ou executar um *javascript* no servidor. É possível adicionar condições para verificar se um alerta deve ser enviado ou se uma ação deve ser executada. Agentes podem ser agendados para serem executados numa determinada frequência e horário e é possível que usuários "assinem" o agente para que possam receber seu conteúdo.

Quando um usuário recebe um alerta na página inicial do OBIEE, o ícone de alerta é destacado como mostrado na figura abaixo.

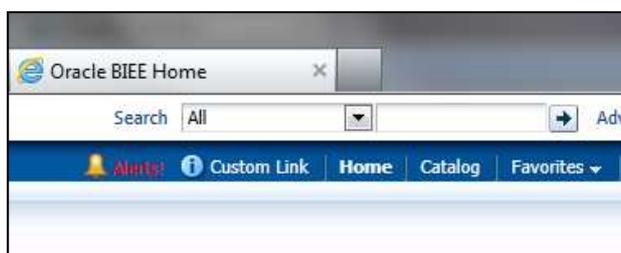


Figura 36: Ícone de alerta destacado

Ao clicar nesse ícone, o usuário pode visualizar o conteúdo do alerta.

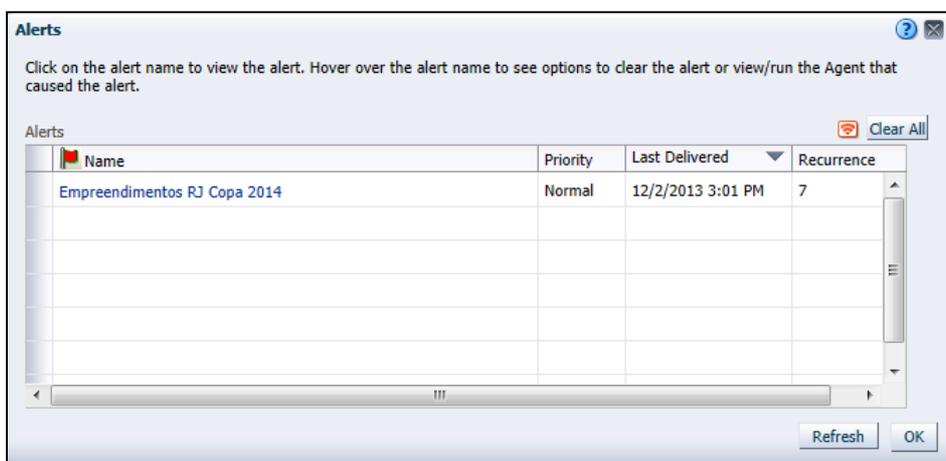


Figura 37: Janela contendo a lista de Alertas

## 5.4 Segurança

A segurança no OBIEE é uma das partes mais complexas da suíte. É possível construir modelos de segurança extremamente sofisticados utilizando os recursos disponibilizados pela ferramenta. A seguir, será apresentada uma visão geral dos objetos utilizados para realizar a segurança e quais níveis de segurança o OBIEE fornece.

Existem três tipos de objetos disponíveis para a organização da segurança no OBIEE: usuários, grupos e *application roles*.

#### **5.4.1 Usuários**

Normalmente é criado um usuário para cada funcionário que irá usar o OBIEE. Por exemplo, se forem necessários 10 usuários leitores de relatórios, 5 usuários autores de relatórios, e 1 administrador, então será necessário criar 16 usuários.

#### **5.4.2 Grupos**

Usualmente é criado um grupo para cada área funcional da empresa. Por padrão, são criados 3 grupos no momento da instalação do OBIEE, BICongomers, BIAuthors, e BIAdministrators, porém é possível criar mais grupos caso seja necessário. Grupos podem conter usuários e grupos.

A criação de usuários e grupos é feita pela web através da página do Oracle WebLogic Server Administration Console. A instalação do ambiente do OBIEE possui um serviço de diretórios embutido. Porém, é possível utilizar outros serviços de diretórios como Oracle Internet Directory ou Microsoft Active Directory.

#### **5.4.3 Application Roles**

As *application roles* podem ser criadas para cada tipo de atividade realizada no ambiente do OBIEE. Por exemplo, pode ser criado uma *application role* para analistas de vendas, e esta role poderá ter permissão para visualizar, editar e criar relatórios relacionados a vendas da empresa. Privilégios podem ser concedidos apenas para *application roles* e usuários.

Um dos objetivos das *application roles* é criar uma camada entre o ambiente do OBIEE e os grupos e usuários do serviço de diretórios. Estes grupos e usuários recebem uma ou mais *application roles*. Desse modo, não é necessário realizar mudanças profundas no serviço de diretório para adequá-lo ao ambiente do OBIEE, pois todos os privilégios podem ser feitos utilizando *application roles* no ambiente do OBIEE.

Por padrão, são criadas 3 *application roles* no momento da instalação do OBIEE:

- BICustomer - esta *role* concede ao usuário acesso à visualização de objetos do catálogo do OBIEE como análises e dashboards, e permite visualizar e agendar relatórios. Essa *role* não permite a criação de novos objetos
- BIAuthor - esta *role* concede os mesmos privilégios da *role* BICustomer e permite a criação de novas análises, dashboards e outros objetos do catálogo do OBIEE
- BIAadministrator - esta *role* concede os mesmos privilégios das *roles* BICustomer e BIAuthor e permite que os usuários administrem todas as partes do ambiente do OBIEE

Application *roles* podem conter grupos, usuários e outras *application roles* e são criadas pela Web na página do Fusion Middleware Control.

#### 5.4.4 Níveis de Segurança

O OBIEE permite que sejam configurados acessos em três níveis diferentes:

- Segurança em nível de linha - Neste nível é possível restringir quais dados os usuários irão acessar nas análises e relatórios. É possível restringir acesso aos dados de todo o conteúdo de uma área de assunto ou apenas aos dados de determinadas tabelas e colunas de uma área de assunto na camada de apresentação. Também é possível filtrar os dados que cada usuário poderá visualizar. Por exemplo, usuários da filial do Rio de Janeiro não podem visualizar dados da filial de São Paulo. Esse nível de segurança é feito utilizando o *client* Administration Tool.
- Segurança em nível de funcionalidade de produto - Neste nível é possível conceder e restringir acesso à determinados produtos e funcionalidades como criar análises, visualizar *dashboards*, e criar relatórios. Este nível de segurança é configurado pela web através da página de administração do OBIEE.
- Segurança em nível de objetos individuais do catálogo - Neste nível é possível conceder e restringir acesso à objetos individuais do catálogo do OBIEE como, por exemplo, uma análise específica, um *dashboard* ou uma pasta. Este nível de segurança é feito utilizando a página de catálogo do OBIEE ou o *client* Catalog Manager.

### 5.4.5 Exemplo de Concessão de Privilégios

Para exemplificar o uso da segurança no OBIEE, foi criado um usuário chamado "usuario1", um grupo chamado "grupo1" e uma *application role* chamada "role1". Para essa *application role*, será concedido acesso apenas aos dados do Rio de Janeiro e apenas à medida "Val Exec Financ".

Primeiramente, através da página do Oracle WebLogic Server Administration Console, o "usuario1" é inserido como membro do "grupo1".

Em seguida, através da página do Fusion Middleware Control, o "grupo1" é inserido na *application role* "role1".

Depois, utilizando o Administration tool, é criada uma variável de sessão que irá conter o valor "Rio de Janeiro". Variáveis de sessão são similares às variáveis de repositório criadas anteriormente neste projeto. Porém, ao contrário das variáveis de repositório, a inicialização de variáveis de sessão não é agendada. A inicialização ocorre quando um usuário inicia uma sessão.

Por fim, também utilizando o Administration tool, a "role1" recebe um filtro que usa a variável de sessão para permitir acesso apenas aos dados do Rio de Janeiro e é removido o acesso às medidas "Val Total Aditivo" e "Val Exec Financ mais Aditivos".

A figura abaixo apresenta uma análise configurada para mostrar todas as medidas da tabela fato e todos os empreendimentos de todas as cidades, porém, como foram adicionados filtros para a "role1", o "usuario1" visualiza apenas os dados dos empreendimentos do Rio de Janeiro e a medida "Val Exec Financ".

Empreendimento▲▼	Ano	Valor Execucao Financeira	Valor Total Aditivo	Val Exec Financ mais Aditivos
BRT Transcarioca (Aeroporto / Penha / Barra)	2011	1,341,226,462		
	2012	25,576,941		
<b>BRT Transcarioca (Aeroporto / Penha / Barra) Total</b>		<b>1,366,803,403</b>		
Implantação de Piers	2011	610,265		
	2013	240,700,774		
<b>Implantação de Piers Total</b>		<b>241,311,039</b>		
Recuperação e revitalização dos sistemas de pistas e pátios	2011	64,558,099		
	2012	3,991,175		
<b>Recuperação e revitalização dos sistemas de pistas e pátios Total</b>		<b>68,549,274</b>		
Reforma do Estádio Mário Filho (Maracanã)	2010	705,589,144		
	2011	23,519,108		
<b>Reforma do Estádio Mário Filho (Maracanã) Total</b>		<b>729,108,252</b>		
Reforma do Terminal de Passageiros 1 e Obras Complementares	2009	30,766,062		
	2010	5,706,843		
	2011	13,430,000		
	2012	160,353,000		
<b>Reforma do Terminal de Passageiros 1 e Obras Complementares Total</b>		<b>210,255,906</b>		
Reforma do Terminal de Passageiros 2	2008	63,440,000		
	2009	12,534,663		
	2011	2,745,790		
	2012	68,550,000		
<b>Reforma do Terminal de Passageiros 2 Total</b>		<b>147,270,454</b>		

Figura 38: Análise filtrada através de segurança

## 6 Desenvolvimento no Pentaho

### 6.1 Conexão com o banco de dados

#### 6.1.1 Pentaho Schema Workbench (PSW)

Antes da criação do repositório, deve-se estabelecer a conexão da ferramenta com o banco de dados em questão. Para o desenvolvimento no Pentaho, foi utilizado o PSW conectado com o PostgreSQL.

No diretório de instalação do Mondrian, deve existir um *driver* JDBC de conexão compatível com a versão do banco de dados utilizado. Após essa garantia, a conexão é configurada como na figura abaixo.

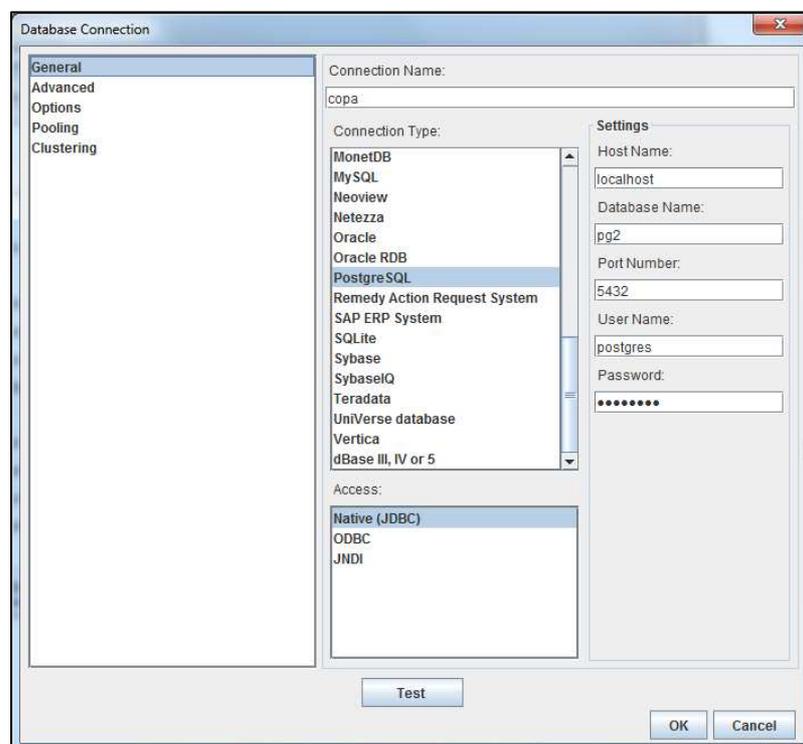


Figura 39 - Definindo parâmetros para conexão no Mondrian

## 6.1.2 Pentaho BI Server

Da mesma forma, para utilização das funcionalidades desta ferramenta do Pentaho, é necessário criar a conexão com o banco de dados. Para esta módulo, a conexão é criada no Pentaho Administration Console.

Também é necessário um *driver* JDBC de conexão compatível com a versão do banco de dados utilizada e a configuração é a seguinte:

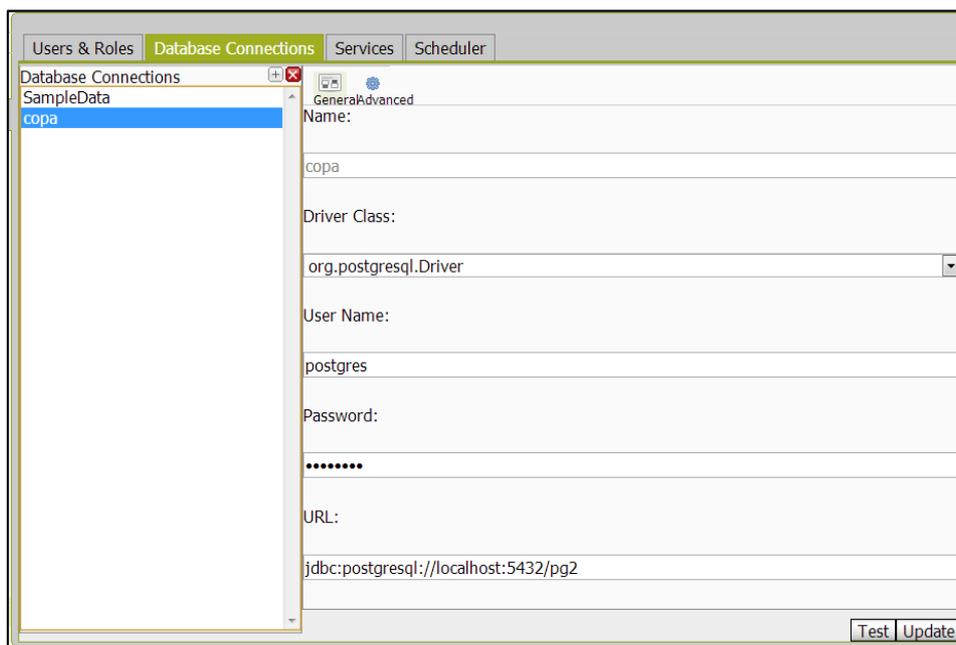


Figura 40 - Definindo parâmetros para conexão no Pentaho BI Server

## 6.2 Criação do repositório

A criação do repositório no Mondrian é feita em etapas. Para este projeto, optou-se pela criação de um único Cubo, pois, como já foi dito anteriormente, o objetivo do trabalho é a comparação entre duas soluções e não uma análise aprofundada dos dados em questão.

### 6.2.1 Criando o cubo

O cubo é criado tendo como base uma tabela fato, que neste caso é referente às execuções financeiras realizadas da Copa. A partir desta tabela, derivamos dimensões que, já mapeadas no modelo dimensional, são úteis para algum tipo de análise. Podem

ser criadas também algumas métricas dependendo do tipo de estudo que deseja ser realizado sobre a base.

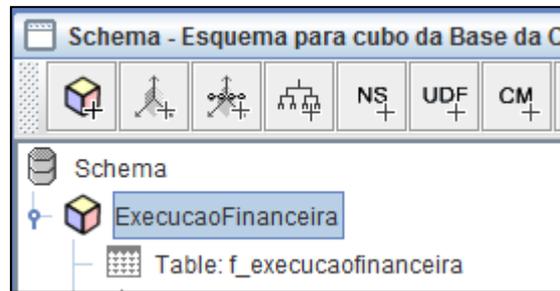


Figura 41 - Adicionando uma tabela Fato ao cubo

### 6.2.2 Criando as dimensões

Após a definição da tabela fato, são criadas as dimensões.

Para o cubo ExecucaoFinanceira, foram criadas as mesmas dimensões do OBIEE.. Ao utilizar estas dimensões, é possível elaborar diversos tipos de análise como, por exemplo, a abertura de Empreendimento por Cidade Sede.

É importante mencionar a especificidade da dimensão Instituição, que exerce dois papéis na tabela Fato, como Instituição Contratada e como Instituição Contratante. Foi necessário criar uma dimensão lógica para cada tipo de Instituição, porém referenciando a mesma tabela (D\_INSTITUICAO).

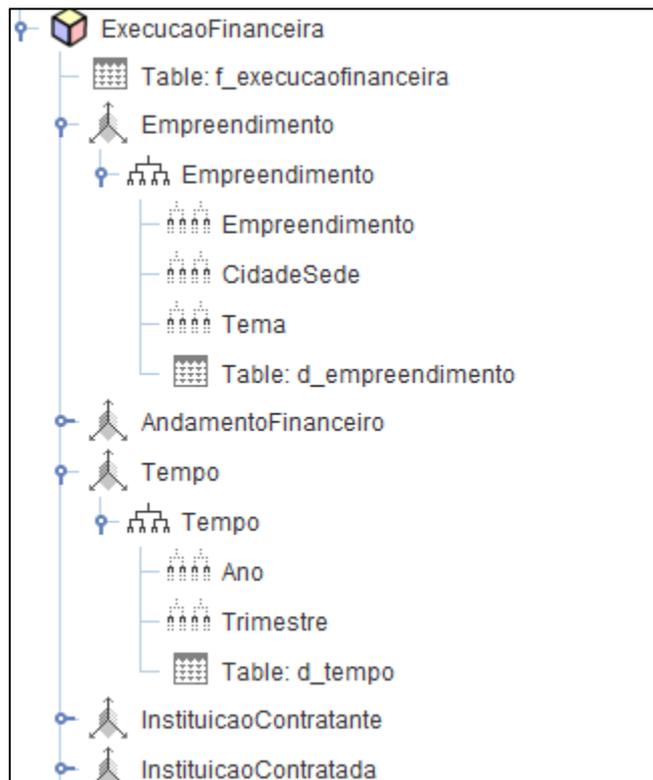


Figura 42 - Adicionando dimensões ao cubo

### 6.2.3 Criando as métricas

Além das dimensões, também são criadas métricas com base nos campos que representam valores na tabela de Fato.

Para a tabela Fato de Execuções Financeiras, existem dois campos (VALEXECFINANC e VALTOTALADITIVO) que remetem a valor e que foram utilizados para a criação das métricas.

Além disso, foram adicionados outros 2 campos que expressam quantidade: QtdAditivo e QtdEmpreendimento. Com estes campos, é possível, por exemplo, obter rapidamente a quantidade de empreendimentos concluídos por Cidade.



Figura 43 - Adicionando as métricas ao cubo

### 6.2.4 Criando medida calculada

No Pentaho, é possível também criar campos calculados a partir de outras métricas já criadas. Esta técnica foi utilizada para a criação da medida ValAditivoValExec, que nada mais é que a soma das medidas ValorTotalAditivo e ValorExecucaoFinanceira.

Pelo lado analítico, este campo identificaria rapidamente o gasto total de um empreendimento, através da soma dos valores de seus aditivos com o valor da execução financeira propriamente dita, além de ser mais um recurso útil disponível na ferramenta.

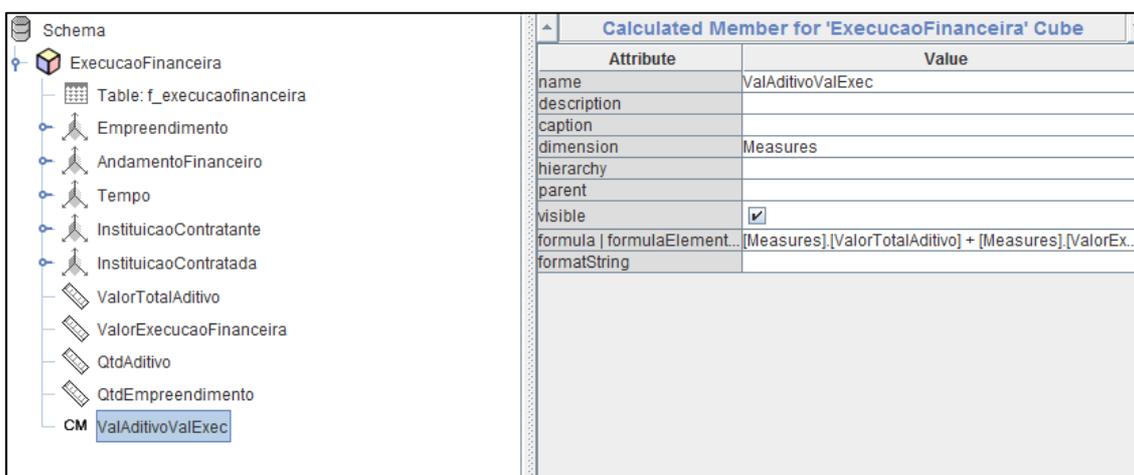


Figura 44 - Adicionando campo calculado

### 6.2.5 Criando medida de série temporal

Foi criada uma medida de série temporal para exemplificar mais um recurso do Pentaho. Neste caso, a year-to-date para o Valor Total de Aditivo que nada mais é do que a soma do valor de aditivos financeiros até uma determinada data. O exemplo abaixo mostra, por ano, a abertura por trimestre junto com seu valor total. Para a medida de série temporal criada, é mostrada a soma dos valores até o trimestre que deseja visualizar.

AndamentoFinanceiro		Measures	
Tempo		● ValorTotalAditivo	● YTD Valor Aditivo
⊕ All AndamentoFinanceiros	⊖ All Tempos	1.165.343.610,09	
	⊕ 2008	34.876.415,86	34.876.415,86
	⊕ 2009	11.793.694,51	11.793.694,51
	⊕ 2010	957.424.665,06	957.424.665,06
	⊖ 2011	123.458.452,9	123.458.452,9
	2011 T1	8.949.779,73	8.949.779,73
	2011 T2	89.613.728,18	98.563.507,91
	2011 T3	21.006.394,66	119.569.902,57
	2011 T4	3.888.550,33	123.458.452,9
	⊖ 2012	29.990.381,76	29.990.381,76
	2012 T1	53.350.303,79	53.350.303,79
	2012 T2	596.090,49	53.946.394,28
	2012 T3	-23.956.012,52	29.990.381,76
	2012 T4	0	29.990.381,76
	⊕ 2013	7.800.000	7.800.000

Figura 45 - Criação de medida calculada

### 6.2.6 Criação de tabela com dados pré-agregados

Utilizou-se também o conceito de tabelas com dados pré-agregados, que aumenta a performance dos cubos OLAP criados no Mondrian.

Embora neste trabalho a massa de dados não seja tão grande a ponto de haver a necessidade de criação destas tabelas, elas foram criadas a fim de termos mais um componente da solução para comparação.

Primeiramente, a conexão à fonte de dados é feita junto com a busca (padrão, no caso do PAD) ao arquivo do esquema Mondrian que foi criado. É interessante notar que, após a definição destes parâmetros, a própria ferramenta sugere quais tabelas podem vir a ser agregadas e mostra o ganho de performance com a agregação proposta. Como dito anteriormente, no caso deste projeto o ganho de desempenho é mínimo.

Feito isso, foi escolhida a agregação de Execuções Financeiras por Cidade Sede apenas para exemplificar. Ao final deste processo, o próprio PAD gera a DDL para criação das tabelas no PostgreSQL e a DML para a população das mesmas.

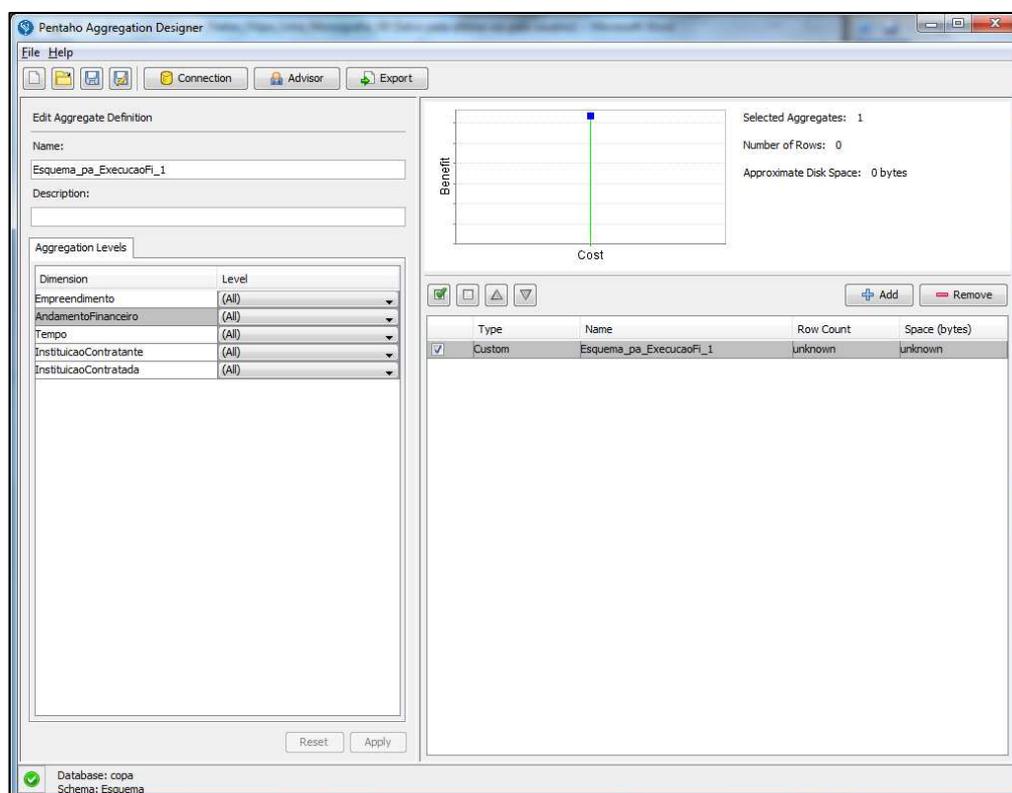


Figura 46 - Desenvolvimento de tabelas agregadas no Aggregation Designer

### 6.2.7 Utilização de variáveis

Podem ser definidos parâmetros, o que simplifica o trabalho no que diz respeito ao uso das Action Sequences.

Com a utilização dos parâmetros, não é mais necessário utilizar as sequências de ações. Claro que, para uma lógica mais complexa, as Action Sequences ainda podem ser utilizadas.

Como foi dito anteriormente, o único problema da utilização de variáveis é que para cada fonte de dados, a implementação é diferente. Enquanto a utilização em consultas padrões é trivial, para utilizar este conceito em linguagem MDX e consultas MQL é um pouco mais complexo. Abaixo, um exemplo simples de utilização de parâmetro em uma consulta padrão SQL.

```
SELECT CAL_YEAR
FROM COPA.D_TEMPO
WHERE TO_CHAR(CALENDAR_DATE, 'DD/MM/YYYY') =
TO_CHAR(SELECT CURRENT_DATE;);
```

Desta forma, o valor da coluna CAL\_YEAR será igual ao ano corrente.

### 6.2.8 Publicando o repositório no servidor

Após a criação do cubo com as suas devidas dimensões e métricas, é necessário publicar o repositório no Pentaho BI Server para que sejam criadas análises e relatórios a partir dele.

Um arquivo XML é criado e salvo em um diretório no Pentaho BI Server. Idealmente, é fundamental salvar uma cópia também na máquina de desenvolvimento porque qualquer alteração posterior poderá ser publicada da mesma forma.

Houve grande dificuldade e diversas tentativas de publicação do cubo após o seu desenvolvimento, sendo necessário instalar uma versão anterior do PSW para que fosse possível publicar.

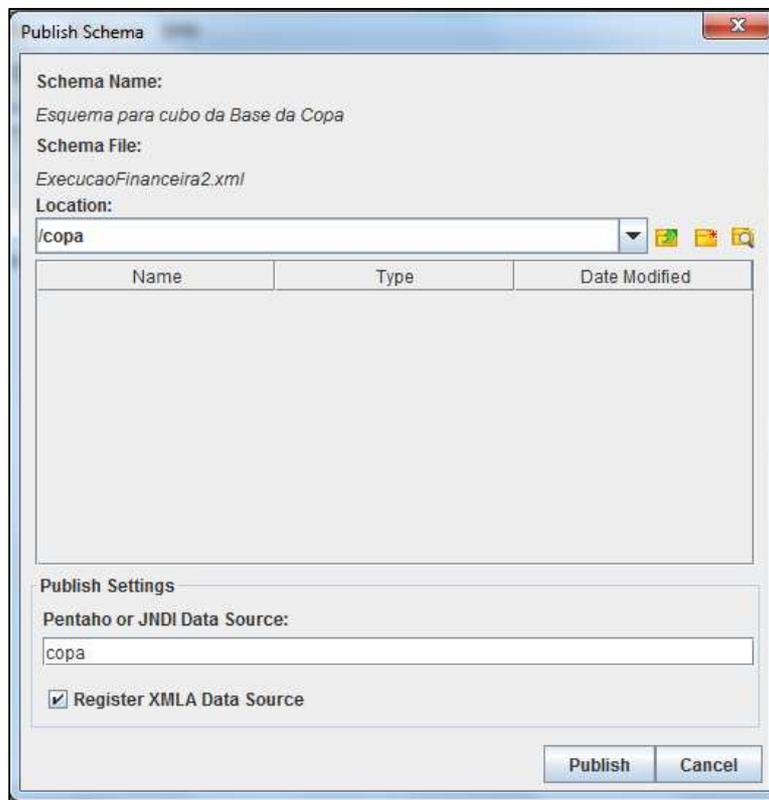


Figura 47 - Publicando o repositório

### **6.3 Criando metamodelo e análises**

Com o cubo criado e publicado com as devidas dimensões e métricas, é chegada a hora da criação das análises. Para tal, deve ser definido um metamodelo de negócios que auxiliará, juntamente com o cubo, na criação das mesmas.

O metamodelo é importante porque nele definimos as colunas que os usuários finais devem ter acesso, com base nas tabelas já mapeadas no modelo dimensional. Além disso, na ferramenta de criação do metamodelo, também é possível definir a quais linhas do relatório os usuários podem ter acesso.

Neste trabalho, o metamodelo foi usado apenas para a criação do relatório no Pentaho Report Designer, que será mostrado mais adiante.

#### **6.3.1 Criação do metamodelo de negócio**

Mais uma funcionalidade da suíte Pentaho é a capacidade de criação de um modelo de metadados de negócio através do Pentaho Metadata Editor. Foram mapeadas as tabelas de dimensão e a tabela Fato modeladas e criadas no PostgreSQL.

Quando carregadas no Metadata Editor, é possível criar relacionamentos, formas de agregação e definir segurança dos metadados. Uma vez criado, é necessário que o metamodelo também seja publicado no repositório do Pentaho BI Server (processo análogo à publicação no PSW).

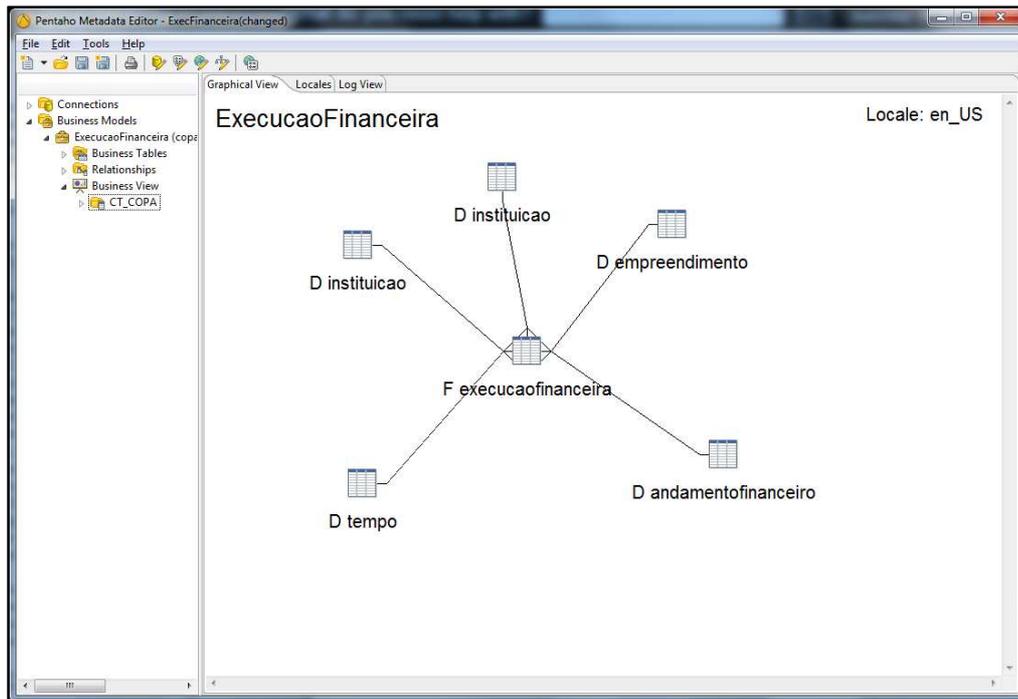


Figura 48 - Criação do metamodelo de negócios

### 6.3.2 Criação de relatório

Para desenvolvimento dos relatórios no Pentaho, optou-se pela criação de relatórios diferentes do que já foram criados no OBIEE. Existem diversas maneiras para criar um relatório utilizando as ferramentas Pentaho. A forma utilizada para exemplificação neste trabalho foi a criação de um relatório no Pentaho Report Designer a partir dos metadados criados pelo Metadata Editor.

No PRD, é possível criar um relatório a partir de um Wizard bem intuitivo que possibilita desde a escolha de um template até a criação da conexão com o banco de dados.

Utilizando o metamodelo criado anteriormente, foi elaborada uma consulta simples que visa retornar o valor total das execuções financeiras por cidade sede. O seu resultado será exibido na forma de um relatório simples, que é exportado automaticamente para um arquivo em pdf, conforme figura abaixo.

The screenshot shows a PDF document titled "report-designer-preview-7077040007937522935.pdf" in Adobe Reader. The report content is as follows:

Valor de Execuções Financeiras por Cidade Sede	
Valor total das execuções financeiras agrupadas por Cidade Sede	
Nacional	615.574.425,339999990000000000
Manaus	879.087.527,660000000000000000
Belo Horizonte	2.197.494.544,210000000000000000
Porto Alegre	828.270.420,560000000000000000
Rio de Janeiro	2.763.298.326,810000000000000000
Natal	598.095.069,089999990000000000
Fortaleza	1.605.404.093,790000000000000000
Brasília	973.619.962,550000100000000000
Salvador	656.579.179,880000000000000000
Curitiba	665.897.918,760000000000000000
Cuiabá	1.945.783.827,640000000000000000
São Paulo	1.231.363.921,070000000000000000
Recife	

Figura 49 - Relatório gerado pelo Pentaho Report Designer

### 6.3.3 Criação de análises

A criação de análises no Pentaho pode ser feita de duas formas: através da própria ferramenta fornecida pelo BI Server ou através de um *plug-in* denominado Saiku Analytics.

No desenvolvimento de análises para este trabalho, o Saiku foi a ferramenta escolhida, pois possui uma interface mais amigável e fácil de manusear.

Primeiro, seleciona-se o cubo que foi criado e publicado no BI Server pelo Mondrian. Irá abrir uma interface com todas as dimensões e métricas criadas (inclusive com o campo calculado a partir de duas medidas). Assim, basta escolher nas abas “Linhas” e “Colunas” as dimensões e métricas que forem convenientes para a análise que se deseja fazer e a consulta irá ser executada automaticamente.

Também é possível adicionar filtros para as dimensões escolhidas na aba “Filtros”.

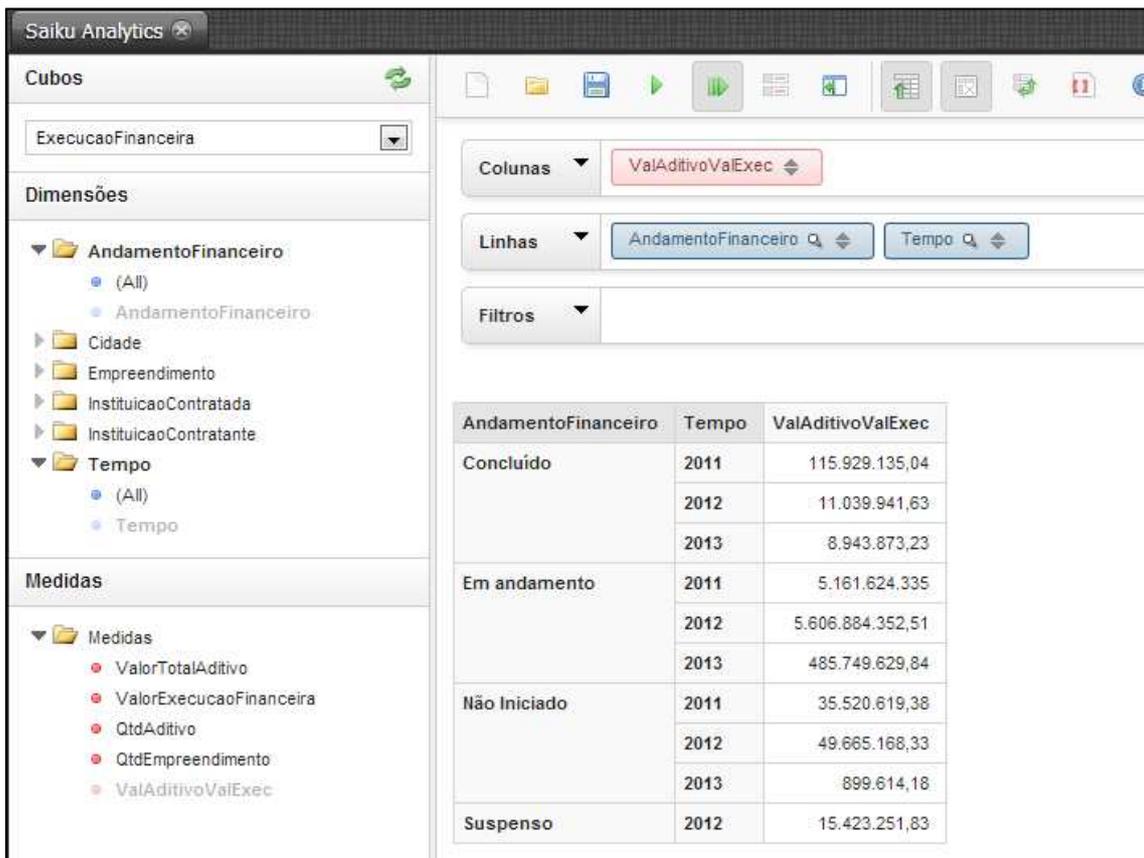


Figura 50 - Análise simples no Saiku Analytics

Para gerar gráficos das análises montadas previamente, o Saiku possui um painel onde é possível alternar a visão de tabela com a visão de gráfico. Existem 10 tipos de gráficos disponíveis para utilização neste *plug-in*.

Análogo ao Report Designer, também é possível exportar relatórios para visualização em planilhas XLS ou arquivos PDF.

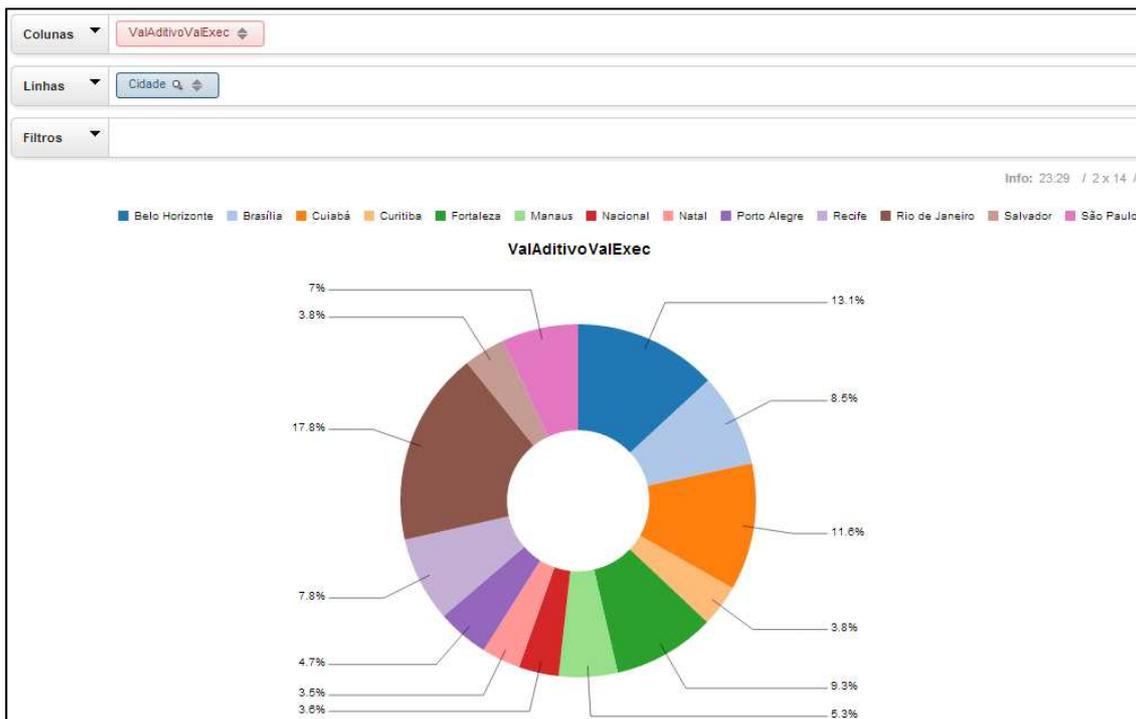


Figura 51 - Análise gráfica no Saiku Analytics

### 6.3.4 Criação de dashboards

Há muita dificuldade para a criação de dashboards no Pentaho Community. Diferentemente de outras ferramentas da solução, os plug-ins CDE/CDF não são “*drag and drop*” e são suscetíveis a muitos erros inesperados. Ao escolher a fonte de dados (seja ela o banco relacional ou o cubo do Mondrian), a aplicação fecha automaticamente.

Além disso, é necessário possuir conhecimento de linguagem HTML e CSS para desenvolver um layout para utilização. Os *templates* presentes nas ferramentas não são intuitivos e claros ao desenvolvedor.

Um ponto positivo seria a criação de data sources a partir do Saiku (um *plug-in* em cima de um *plug-in*). Também destaca-se a rapidez no retorno de resultado, muito pelo tamanho da base de dados e dependendo também da forma com que a conexão foi criada.

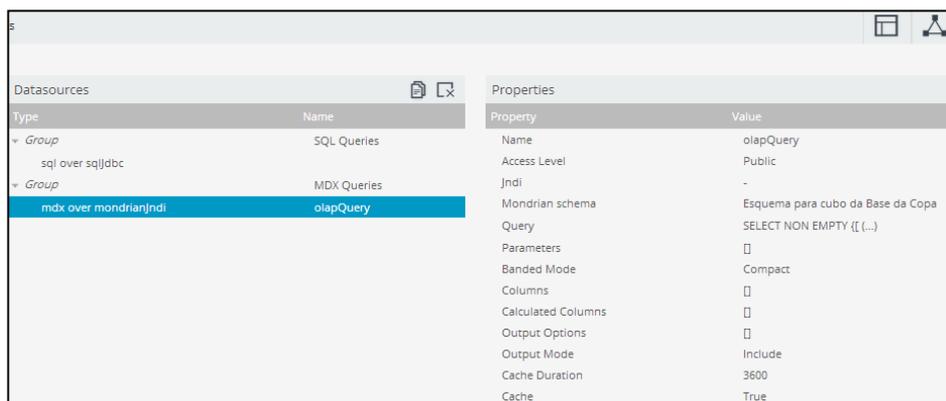


Figura 52 – Exemplo de criação de dashboard no Pentaho Community

## 6.4 Alertas e ações

Após toda a criação do cubo no Mondrian, foram criadas algumas consultas pré-definidas sobre o esquema construído. O componente que nos auxiliou nesta tarefa foi o Pentaho Design Studio.

Foram criadas as chamadas Action Sequences, que como o próprio nome diz, são sequências de ações ordenadas que executam tarefas únicas como: gerar uma consulta e, a partir dessa consulta, produzir um gráfico e enviar o relatório final por e-mail. Além disso, uma action sequence pode ser capaz de invocar outras e executar outros componentes.

Assim como os metadados e o cubo, o arquivo .xaction que é gerado ao final do desenvolvimento no PDS deve ser publicado no repositório da solução. Também houve uma certa dificuldade no momento da publicação do arquivo no servidor.

Abaixo, a tela de criação da consulta que servirá de base para a criação dos gráficos que serão enviados por e-mail.

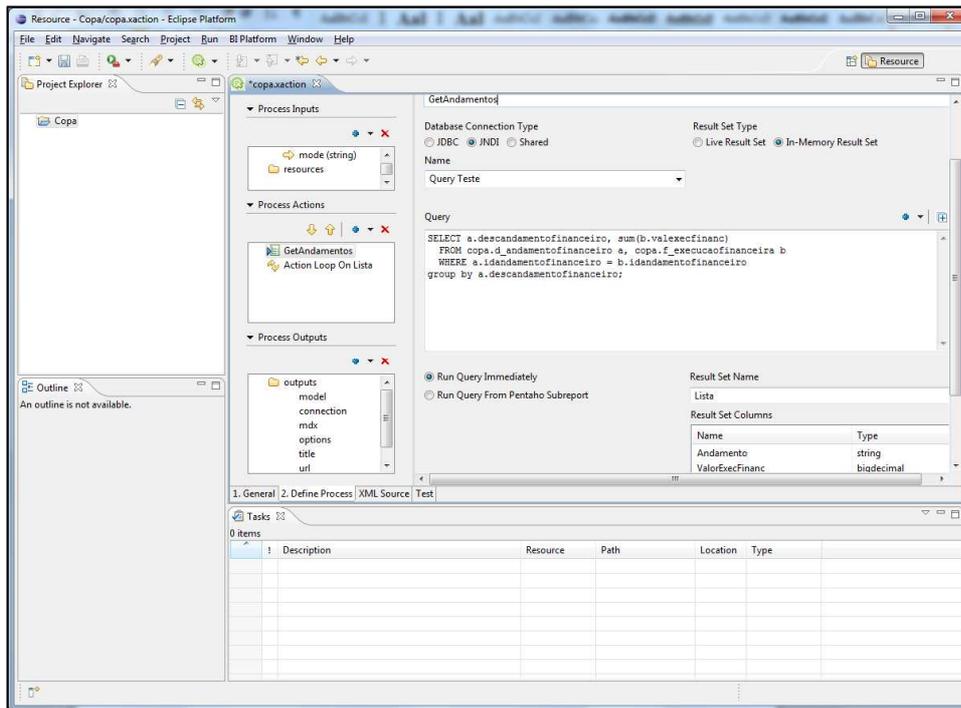


Figura 53 - Criação de Action Sequences no Pentaho Design Studio

## 6.5 Segurança

### 6.5.1 Definindo configurações de segurança

Primeiramente, deve-se buscar as informações de cada papel e usuário presente no servidor para então poder definir o nível de acesso de cada um deles. Isto pode ser feito tanto definindo a URL do servidor para conexão e obtenção das informações (exemplo abaixo) quanto localizando um arquivo onde estão presentes estas definições.

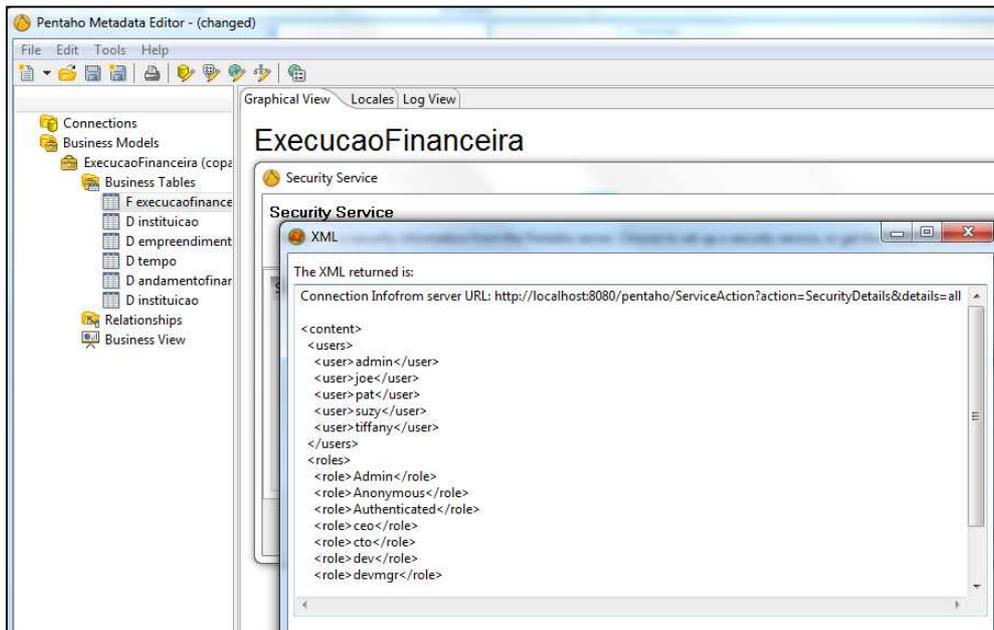


Figura 54 - Definições de papéis e usuários no Pentaho BI Server

### 6.5.2 Alterando restrições de segurança

Com as definições atuais carregadas, chega a hora de alterar as restrições existentes. No Pentaho Metadata Editor, isto pode ser feito clicando na tabela que se deseja modificar e alterando a permissão para um determinado papel ou usuário.

É possível alterar permissões para o modelo, a tabela ou somente a coluna. Deste modo, por exemplo, um usuário pode não conseguir visualizar uma determinada coluna de um relatório específico.

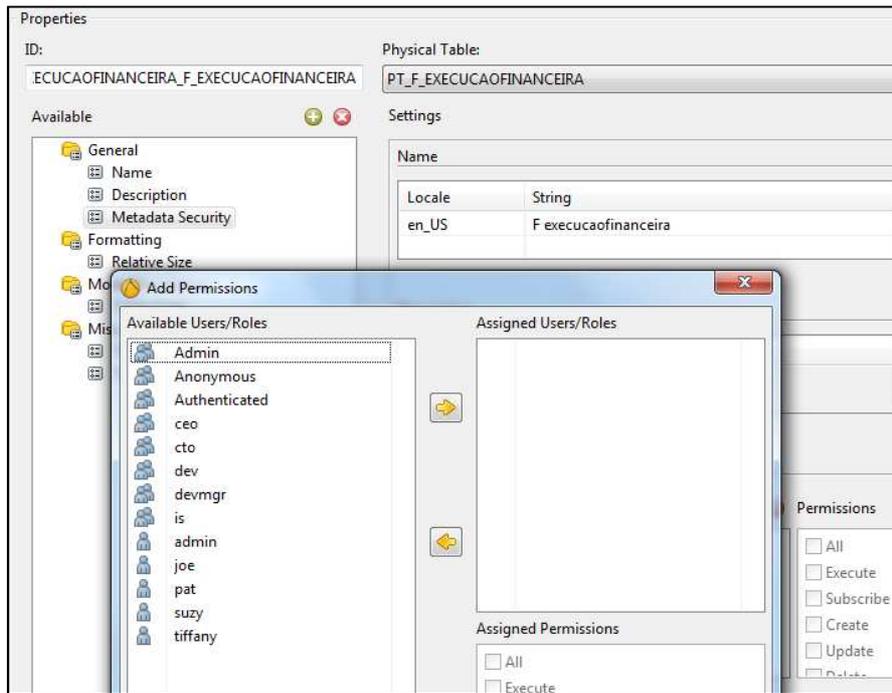


Figura 55 - Alteração de permissão de segurança nos metadados

Para alteração de permissões de acesso a cubos do Mondrian, é necessário alterar o próprio arquivo XML gerado, definindo quais papéis e/ou usuários terão acesso a determinadas métricas e dimensões.

## 7 Comparação do Desenvolvimento no OBIEE e no Pentaho

### 7.1 Criação de repositório

O banco de dados utilizado pelo Pentaho foi o PostgreSQL. Para o OBIEE, foi utilizado o Oracle Database.

#### Conexão ao Banco de Dados

A conexão dos componentes do Pentaho com o PostgreSQL foi trivial. Bastou copiar o *driver* JDBC do PostgreSQL para a pasta da ferramenta em questão e seguir as instruções de conexão.

No OBIEE, foi encontrada dificuldade para estabelecer conexão entre o OBIEE e o Oracle Database. Para conectar ao Oracle DB foi utilizado o Oracle Call Interface (OCI). Para esse tipo de conexão, é necessário inserir uma URL para abrir a conexão com o banco. A conexão não era estabelecida porque a URL fornecida estava incorreta. Após muitas pesquisas, tentativas e erros, foi encontrada a URL correta e a conexão foi estabelecida utilizando OCI.

#### Criação do Metamodelo

No Pentaho, para criação do metamodelo de negócios foi utilizado o Metadata Editor, enquanto para a criação do cubo foi utilizado o Mondrian OLAP Server. Diferentemente do OBIEE onde a criação dos dois ocorre na mesma plataforma.

Em ambos componentes do Pentaho foram encontradas dificuldades para publicação do repositório no servidor. No caso do Mondrian, houve a necessidade de instalação de uma versão anterior à que estava sendo utilizada para que a publicação pudesse ser feita. Após algumas pesquisas para encontrar o erro, foi descoberto que a URL onde deveria ser publicado estava incorreta. Um ponto negativo deste trabalho foi

a ausência dessas informações no *log*, o que facilitaria a descoberta e agilizaria o desenvolvimento.

Uma facilidade encontrada no desenvolvimento do repositório do OBIEE foi que o mapeamento da camada física e criação de relacionamentos são feitos graficamente. Isso acelera o aprendizado da ferramenta e evita que erros sejam cometidos logo no início do desenvolvimento da aplicação.

Outra funcionalidade que facilitou o desenvolvimento da aplicação no OBIEE foi a verificação de consistência do repositório no Administration Tool. Esta funcionalidade é bastante útil, pois verifica, informa e, quando possível, corrige automaticamente possíveis inconsistências e erros presentes no repositório de metadados do OBIEE. Isso reduz as chances de que o repositório seja disponibilizado com problemas ao usuário final.

Um ponto negativo encontrado e que diminui a eficiência no desenvolvimento no OBIEE é que, após as mudanças no conjunto de metadados do repositório do OBIEE, é necessário realizar um *upload* do arquivo do repositório no servidor através de um portal web e, em seguida, reiniciar os serviços do OBIEE para que as mudanças sejam efetivadas.

## **7.2 Medidas Calculadas**

A criação de medidas calculadas no Pentaho foi bastante simples. De um modo geral, a implementação é a mesma da criação de uma métrica comum. A única diferença é que deve ser adicionada uma expressão no campo “Formula”, também em linguagem MDX.

No OBIEE, a criação de medidas calculadas é semelhante à criação no Pentaho. A diferença é que ao invés de utilizar MDX para criar a medida, são utilizadas funções do próprio OBIEE.

As duas plataformas permitem que sejam criadas novas medidas no cubo/repositório de dados quanto em um determinado relatório.

### **7.3 Criação de Hierarquias**

A criação de hierarquias no Pentaho seguiu a lógica da criação da dimensão e ocorreu sem maiores problemas. Ao criar a dimensão, hierarquia e níveis a própria ferramenta indica se existe algum erro no momento da criação e não permite a publicação do esquema caso seja encontrado algum. Desta forma, a interface para a criação agiliza o desenvolvimento.

Além disso, é possível definir parâmetros para as hierarquias como, por exemplo, seu membro *default* e se ela será visível ou não.

A criação de hierarquias no OBIEE foi um pouco mais complicada. O processo de criação não é direto e rápido. É necessário configurar vários parâmetros nos membros da hierarquia, nas dimensões e nos fatos do modelo.

### **7.4 Dados Pré-Agregados**

O Pentaho permite a criação de dados pré-agregados utilizando o Pentaho Aggregation Designer. Um ponto positivo no PAD é a indicação de qual tabela é uma candidata a agregação, de acordo com estatísticas da própria ferramenta.

O OBIEE permite configurar a funcionalidade de dados pré-agregados de maneira gráfica facilitando o uso dessa funcionalidade.

### **7.5 Utilização de Variáveis/Parâmetros**

O Pentaho permite a criação de variáveis, que são chamadas parâmetros. Porém, o uso de parâmetros no PRD não é trivial. O principal problema é que para cada fonte de dados a definição é diferente. A utilização de parâmetros com consultas padrões em banco de dados é fácil, porém, a utilização com consultas MDX, MQL e outras fontes de dados é um tópico muito mais avançado.

O OBIEE também permite a utilização de variáveis para facilitar atividades administrativas no ambiente. Por exemplo, pode ser utilizada uma consulta SQL ao banco de dados para configurar o valor de uma variável.

## 7.6 Medidas de Séries Temporais

No OBIEE, a criação de medidas de séries temporais é feita utilizando funções do OBIEE. No Pentaho, o campo de medida é criado utilizando expressão em linguagem MDX. Nas duas plataformas, é possível criar medidas de séries temporais tanto na estrutura de metadados do modelo quanto apenas em um determinado relatório.

## 7.7 Análises, Relatórios e Dashboards

### Análises

É possível criar análises utilizando a ferramenta nativa do Pentaho, que tem como base a biblioteca JPivot. Porém, esta biblioteca não possui uma interface amigável e poucas funcionalidades além de ocorrerem alguns erros e avisos durante a criação das análises. Por isso foi utilizado o *plug-in* Saiku Analytics, que também permite a criação de análises com mais funcionalidades e melhor resultado final.

No OBIEE não há necessidade de instalar *plug-ins* extras para criar análises, pois o mesmo já possui as ferramentas necessárias. Porém, sua interface pode ser confusa devido à grande quantidade de informação apresentada na tela de criação de análises.

No ambiente web para criação de análises no OBIEE, algumas vezes ocorreu o travamento da interface gráfica, sendo necessário fechar e abrir a página web, porém esse problema não aconteceu constantemente. Outro problema que aconteceu raramente foi que o Internet Explorer consumia muita memória após permanecer muito tempo aberto. Esse problema pode ser do OBIEE ou do próprio Internet Explorer.

### Relatórios

Para criação de relatórios no Pentaho, podemos utilizar tanto a função “New Report” no próprio BI Server quanto a ferramenta Pentaho Report Designer. No entanto, as duas ferramentas acessam o modelo criado no Metadata Editor, não sendo possível utilizar o repositório criado no Mondrian.

No OBIEE, a ferramenta de criação de relatórios pode se conectar facilmente à área de assunto criada no repositório do OBIEE como também a outras fontes de dados. Em geral, o desenvolvimento de relatórios ocorreu sem problemas.

### Dashboard

A maior dificuldade encontrada em todo o desenvolvimento no Pentaho foi na criação dos *dashboards*. Para criação dos mesmos, é necessário instalar dois *plug-ins* no Pentaho BI Platform: o Pentaho Community Dashboard Editor (CDE) e o Pentaho Community Dashboard Framework (CDF). A criação em si acontece no Pentaho CDE.

A interface deste componente é bem confusa. Existem apenas dois *templates* de *dashboard* disponíveis e mesmo assim a ferramenta não apresenta uma maneira clara para desenvolver em cima destes *templates*.

Ao criar um layout, é preciso saber as linguagens HTML e CSS, o que faz o desenvolvimento ficar mais trabalhoso. Como dito na seção de desenvolvimento, a ferramenta para criação dos *dashboards* deixa muito a desejar quando comparada à outras ferramentas da solução Pentaho como um todo.

Para finalizar, ao escolher os modos de assistente de criação a ferramenta trava, sendo preciso reinicializar o servidor do Pentaho. Ocorre o mesmo quando é preciso incluir algum *data source*.

A criação de *dashboards* no OBIEE é bem mais simples do que a criação no Pentaho e do que a criação de análises e relatórios no próprio OBIEE. Utilizando uma interface gráfica, o usuário constrói o *dashboard* com *drag and drop*, adicionando objetos como análises, previamente criados. Além disso, é possível utilizar parâmetros, variáveis e filtrar o conteúdo do *dashboard* dinamicamente.

## 7.8 Alertas e Ações

Através da ferramenta Pentaho Design Studio, o desenvolvimento de ações foi simples e objetivo. Através do *wizard* que é fornecido por este componente foi possível sem qualquer tipo de problema criar uma sequência de ações que realiza uma consulta, gera um gráfico e envia este gráfico por e-mail.

A criação e agendamento de alertas é bastante simples no OBIEE. É possível enviar alertas aos usuários finais através de diferentes meios como e-mail ou aviso no portal web do OBIEE. Os alertas podem conter análises e *dashboards*. O OBIEE também permite que sejam executadas diferentes ações como execução de script no servidor ou iniciar um *Web Service*.

## 7.9 Segurança

O Pentaho permite definir parâmetros de segurança. Um ponto positivo do Pentaho neste quesito é que a segurança é tratada diretamente no Metadata Editor e para o cubo em um arquivo XML, sem a necessidade de navegação entre outros ambientes.

No Metadata Editor, basta “apontar” para onde estão as definições de segurança do Pentaho BI Server e as mesmas serão carregadas para seus metadados. Após este passo, é possível editá-las tanto para o modelo como um todo quanto para as tabelas e colunas.

A arquitetura da segurança no OBIEE é de difícil entendimento. Podem ser utilizados usuários, grupos e roles, o que pode trazer complexidade desnecessária ao modelo de segurança do ambiente. Além disso, é necessário acessar diversos ambientes diferentes para criar as definições de segurança.

Porém, apesar de complexo, é possível definir um modelo de segurança bastante robusto e personalizado no OBIEE, com integração a serviços de diretórios previamente existentes na empresa atendendo aos diferentes modelos de segurança de cada organização.

## 7.10 Outras Observações

A Oracle disponibiliza diversos tutoriais sobre o OBIEE, além de toda a documentação e acesso livre ao fórum da Oracle. É muito fácil achar informação sobre o OBIEE e também sobre as outras ferramentas da Oracle.

É possível baixar livremente quase todas as ferramentas da Oracle individualmente para uso não comercial. Além disso, são disponibilizadas máquinas virtuais com várias ferramentas instaladas, prontas para desenvolvimento.

Para o Pentaho, é possível encontrar alguns tutoriais pela internet e existem algumas comunidades de desenvolvimento. Também é possível encontrar uma breve descrição dos plug-ins e componentes que estão em pauta para desenvolvimento.

## 8 Conclusões

### 8.1 Considerações Finais

A implementação de uma solução de BI deve estar sempre de acordo com as expectativas e interesses da empresa/instituição que deseja realizar as análises e tomar as decisões, procurando atingir as metas definidas pela mesma.

Contudo, mesmo que a organização tenha acesso a ferramentas adequadas para o tratamento das informações, devido à sua curva de aprendizado, caso a mesma não saiba utilizá-las corretamente, estas informações não terão nenhuma utilidade. Para que as informações oriundas de BI atendam às expectativas esperadas, é preciso que os usuários atendam suas necessidades a partir da análise. Tal análise visa a utilização das informações de BI para tomadas de decisões precisas e coerentes.

Além da tomada de decisão, outros benefícios são notórios com a utilização do BI nas organizações, tais como a facilidade na identificação dos riscos, compreensão das tendências dos negócios e um planejamento corporativo mais amplo. O acesso e a distribuição das informações para envolvimento de áreas estratégicas da empresa são os pilares para obtermos tais benefícios.

Ao final deste trabalho, foi possível entender melhor as características de cada solução desenvolvida. O Pentaho, apesar de ser uma plataforma *open-source*, possui diversas funcionalidades que o OBIEE também fornece, porém com custo inicial menor, sendo mais atrativo para pequenas e até mesmo médias empresas, assim como para organizações não governamentais sem fins lucrativos. O OBIEE fornece um conjunto de ferramentas mais robusto porém com um custo inicial maior, sendo mais voltado para grandes empresas e organizações. Embora ofereça muitas funcionalidades, o desenvolvimento no Pentaho foi mais difícil que no OBIEE porque ao longo da implementação, muitas vezes, não foi possível realizar configurações utilizando uma

interface gráfica, sendo necessário alterar arquivos XML, realizar consultas MDX, entre outros detalhes tecnológicos. Já no OBIEE, são utilizadas interfaces gráficas ao longo de todo o processo de desenvolvimento. Desta forma, concluímos que o OBIEE é superior ao Pentaho no que diz respeito à facilidade de implementação da solução.

A aplicação das métricas do Gartner Group possibilitou um estudo objetivo das duas plataformas de BI escolhidas, resultando em uma comparação geral que pode ser aplicada a outras plataformas. Assim como as características comparadas no passo-a-passo do desenvolvimento, o contexto das comparações realizadas se referiu a diferentes pontos de vista, seja da organização como um todo ou de sua área de TI, seja do usuário final, dependendo de cada aspecto analisado.

## **8.2 Trabalhos Futuros**

Uma sugestão de trabalho futuro é a continuação de comparação com outras plataformas de Business Intelligence com forte participação no mercado, tendo em vista a existência de diversas plataformas com diferentes características e funcionalidades.

Também seria interessante observar o comportamento das duas plataformas aqui comparadas utilizando grandes bases de dados e elaborando análises mais robustas a fim de comparar também a performance nestes casos, já que este item não foi avaliado neste trabalho.

No que tange a dados abertos governamentais, o presente trabalho pode ser aplicado a diversas outras bases de dados existentes no Portal da Transparência e outros sítios de Governos das diferentes esferas, como demonstrou a prova de conceito apresentada.

## Bibliografias Consultadas

Kimball, Ralph; Ross, Margy. The Data Warehouse Toolkit. John Wiley, 2002.

<http://www.uniriotec.br/~tanaka/SAIN/01-VisaoGeral.pdf>, Acessado em 31 de março de 2013.

Inmon, William. Building the Data Warehouse. Wiley Publishing, 2005.

[http://en.wikipedia.org/wiki/Data\\_warehouse](http://en.wikipedia.org/wiki/Data_warehouse), Acessado em 31 de Março de 2013.

[http://en.wikipedia.org/wiki/Snowflake\\_schema](http://en.wikipedia.org/wiki/Snowflake_schema), Acessado em 1 de Abril de 2013

<http://www.oracle.com/technetwork/middleware/bi-foundation/bi-foundation/enterprise-edition-platform-compone-093074.html>, Acessado em 24 de outubro de 2013

<http://www.oracle.com/us/bi-enterprise-edition-plus-ds-078848.pdf>, Acessado em 24 de outubro de 2013

[http://docs.oracle.com/cd/E23943\\_01/bi.1111/e22257/toc.htm](http://docs.oracle.com/cd/E23943_01/bi.1111/e22257/toc.htm), Acessado em 24 de outubro de 2013

<http://everythingoracle.com/obieehyp.htm>, Acessado em 26 de outubro de 2013

<http://www.oracle.com/technetwork/middleware/bi-foundation/index-101307.html>, Acessado em 26 de outubro de 2013

<http://www.oracle.com/technetwork/middleware/bi-foundation/financial-reporting-095356.html>, Acessado em 26 de outubro de 2013

<http://oracle-bi.siebelunleashed.com/articles/obiee-architecture/>, Acessado em 27 de outubro de 2013

<http://oracle-bi.siebelunleashed.com/articles/insight-of-obiee-architecture-ii/>, Acessado em 27 de outubro de 2013

[http://gerardnico.com/wiki/dat/obiee/presentation\\_service](http://gerardnico.com/wiki/dat/obiee/presentation_service), Acessado em 27 de outubro de 2013

[http://docs.oracle.com/cd/E14571\\_01/bi.1111/e10540/sqlref.htm](http://docs.oracle.com/cd/E14571_01/bi.1111/e10540/sqlref.htm), Acessado em 27 de outubro 2013

[http://docs.oracle.com/cd/E23943\\_01/bi.1111/e10541/prescatadmin.htm](http://docs.oracle.com/cd/E23943_01/bi.1111/e10541/prescatadmin.htm), Acessado em 27 de outubro de 2013

[http://docs.oracle.com/cd/E23943\\_01/bi.1111/e10540/intro.htm#CIHEDEGE](http://docs.oracle.com/cd/E23943_01/bi.1111/e10540/intro.htm#CIHEDEGE), Acessado em 29 de outubro de 2013

<http://oracle-bi.siebelunleashed.com/articles/insight-obiee-arch-iii/>, Acessado em 29 de outubro de 2013

[http://docs.oracle.com/cd/E14571\\_01/bi.1111/e10541/schedconcepts.htm](http://docs.oracle.com/cd/E14571_01/bi.1111/e10541/schedconcepts.htm), Acessado em 02 de novembro de 2013

<http://oracle-bi.siebelunleashed.com/articles/insight-of-obiee-architecture-iv/>, Acessado em 02 de novembro de 2013

[http://en.wikipedia.org/wiki/Time\\_series](http://en.wikipedia.org/wiki/Time_series), Acessado em 30 de novembro de 2013

[http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/bi11115/biadmin11g\\_02/biadmin11g.htm](http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/bi11115/biadmin11g_02/biadmin11g.htm), Acessado em 29 de novembro de 2013

<http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/bi11117/ps/ps.html?cid=8113&ssid=0>, Acessado em 01 de dezembro de 2013

[http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/bip/bip11g/gettings\\_tarted11116/gs.htm?cid=6229&ssid=0](http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/bip/bip11g/gettings_tarted11116/gs.htm?cid=6229&ssid=0), Acessado em 02 de dezembro de 2013

<http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/bi11115/delivers/alerts.htm?cid=5999&ssid=0>, Acessado em 02 de dezembro de 2013

[http://docs.oracle.com/cd/E28280\\_01/bi.1111/e10543/intro.htm#CJHHHJCG](http://docs.oracle.com/cd/E28280_01/bi.1111/e10543/intro.htm#CJHHHJCG), Acessado em 02 de dezembro de 2013

[http://docs.oracle.com/cd/E28280\\_01/bi.1111/e10543/authentication.htm#CACDACEJ](http://docs.oracle.com/cd/E28280_01/bi.1111/e10543/authentication.htm#CACDACEJ), Acessado em 02 de dezembro de 2013

<http://www.rittmanmead.com/2012/03/an-obiee-11g-security-primer-introduction/>, Acessado em 02 de dezembro de 2013

<http://www.rittmanmead.com/2012/03/obiee-11g-security-week-row-level-security/>, Acessado em 02 de dezembro de 2013

<http://www.rittmanmead.com/2012/03/obiee-11g-security-week-subject-area-catalog-and-functional-area-security-2/>, Acessado em 02 de dezembro de 2013

<http://www.rittmanmead.com/2012/03/obiee-11g-security-week-managing-application-roles-and-policies-and-managing-security-migrations-and-deployments/>, Acessado em 02 de dezembro de 2013

[http://www.peakindicators.com/media\\_pi/Knowledge/oracle%20bi%2011g%20security%20-%20it%20is%20as%20easy%20as%201-2-3.pdf](http://www.peakindicators.com/media_pi/Knowledge/oracle%20bi%2011g%20security%20-%20it%20is%20as%20easy%20as%201-2-3.pdf), Acessado em 03 de dezembro de 2013

<http://www.portaltransparencia.gov.br/copa2014/dados/download.seam>, Acessado em 01 de dezembro de 2013

<http://www.portaltransparencia.gov.br/copa2014/saibamais.seam?textoIdTexto=1>, Acessado em 01 de dezembro de 2013

<http://mondrian.pentaho.com/documentation/faq.php>, Acessado em 9 de novembro de 2013

<http://mondrian.pentaho.com/documentation/olap.php>, Acessado em 9 de novembro de 2013

<http://meteorite.bi/saiku>, Acessado em 9 de novembro de 2013

<http://wiki.pentaho.com/display/COM/Pentaho+Data+Integration+Articles>, Acessado 9 de novembro de 2013

<http://wiki.pentaho.com/display/ServerDoc2x/BI+Server+2.x-3.x+Community+Documentation>, Acessado em 19 de novembro de 2013

<http://wiki.pentaho.com/display/ServerDoc2x/01.+Metadata+Business+Model+Overview>, Acessado em 19 de novembro de 2013

<http://rpbouman.blogspot.com.br/2006/06/pentaho-data-integration-kettle-turns.html>, Acessado em 19 de novembro de 2013

<http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+Screenshots>, Acessado em 19 de novembro de 2013

[http://infocenter.pentaho.com/help/index.jsp?topic=%2Faggregation\\_designer\\_guide%2Fconcept\\_pad\\_overview.html](http://infocenter.pentaho.com/help/index.jsp?topic=%2Faggregation_designer_guide%2Fconcept_pad_overview.html), Acessado em 3 de dezembro de 2013

<http://wiki.bizcubed.com.au/xwiki/bin/view/Pentaho+Tutorial/Mondrian+Security>, Acessado em 3 de dezembro de 2013

<http://www.managementsignals.com/mondrian-role-security>, Acessado em 3 de dezembro de 2013

<http://diethardsteiner.blogspot.com.br/2009/11/using-parameters-in-pentaho-report.html>, Acessado em 5 de dezembro de 2013

<http://wiki.pentaho.com/display/ServerDoc2x/Introduction+to+the+BI+Platform>, Acessado em 10 de dezembro de 2013

<http://wiki.pentaho.com/display/ServerDoc2x/Pentaho+Metadata+Editor>, Acessado em 10 de dezembro de 2013

<http://wiki.pentaho.com/display/ServerDoc2x/Design+Studio>, Acessado em 10 de dezembro de 2013

<http://wiki.pentaho.com/display/ServerDoc2x/.01+Introduction>, Acessado em 10 de dezembro de 2013

Pedroso, L., Tanaka, A. and Cappelli, C. (2013) "A Lei de Acesso à Informação Brasileira e os Desafios Tecnológicos dos Dados Abertos Governamentais", <http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2013/0048.pdf>, May.

Azevedo, L., Romeiro, C., Cappelli, C. and Baião, F. (2010), "Passo-a-Passo para Avaliação de Software para Aquisição em Modelos de Processos de Negócio", <http://www.seer.unirio.br/index.php/monografiasppgi/article/view/978/743>, September.

## Anexo I DDL para as Tabelas de Dimensão e Fato

### DDL usada no Oracle Database

```
-----  
-- DDL for Table D_ANDAMENTOFINANCEIRO  
-----  
  
CREATE TABLE "D_ANDAMENTOFINANCEIRO"  
(  "IDANDAMENTOFINANCEIRO" NUMBER,  
  "DESCANDAMENTOFINANCEIRO" VARCHAR2(255 BYTE)  
)  
/  
-----  
-- DDL for Table D_EMPREENDIMENTO  
-----  
  
CREATE TABLE "D_EMPREENDIMENTO"  
(  "IDEMPREENDIMENTO" NUMBER,  
  "IDCIDADESEDE" NUMBER,  
  "DESCEMPREENDIMENTO" VARCHAR2(4000 BYTE),  
  "DESCOBSERVACAO" VARCHAR2(4000 BYTE),  
  "TXTEXPLICATIVOINVESTIMENTO" VARCHAR2(4000 BYTE),  
  "DESCCIDADESEDE" VARCHAR2(255 BYTE),  
  "IDTEMA" NUMBER,  
  "DESCTEMA" VARCHAR2(255 BYTE)  
)  
/  
-----  
-- DDL for Table D_INSTITUICAO  
-----  
  
CREATE TABLE "D_INSTITUICAO"  
(  "IDINSTITUICAO" NUMBER,  
  "IDTIPOINSTITUICAO" NUMBER,  
  "NOMINSTITUICAO" VARCHAR2(255 BYTE),  
  "DESCTIPOINSTITUICAO" VARCHAR2(255 BYTE),  
  "ESFERATIPOINSTITUICAO" VARCHAR2(255 BYTE)  
)  
/  
-----  
-- DDL for Table D_TEMPO  
-----  
  
CREATE TABLE "D_TEMPO"  
(  "CALENDAR_DATE" DATE,  
  "DAY_KEY" NUMBER,  
  "CAL_HALF" NUMBER,  
  "CAL_MONTH" NUMBER,
```

```

"CAL_QTR" NUMBER,
"CAL_YEAR" VARCHAR2(200 BYTE),
"DAY_OF_MONTH" NUMBER,
"PER_NAME_HALF" VARCHAR2(255 CHAR),
"PER_NAME_MONTH" VARCHAR2(255 CHAR),
"PER_NAME_QTR" VARCHAR2(255 CHAR)
)
/
-----
-- DDL for Table D_TEMPO_QTR_GRAIN
-----

CREATE TABLE "D_TEMPO_QTR_GRAIN"
(
  "CAL_HALF" NUMBER,
  "CAL_QTR" NUMBER,
  "CAL_YEAR" VARCHAR2(200 BYTE),
  "PER_NAME_HALF" VARCHAR2(200 BYTE),
  "PER_NAME_QTR" VARCHAR2(200 BYTE),
  "QTR_KEY" NUMBER
)
/
-----
-- DDL for Table F_EXECUCAOFINANCEIRA
-----

CREATE TABLE "F_EXECUCAOFINANCEIRA"
(
  "IDEXECUCAOFINANCEIRA" NUMBER,
  "IDEMPREENDIMENTO" NUMBER,
  "IDANDAMENTOFINANCEIRO" NUMBER,
  "IDINSTITUICAOCONTRATANTE" NUMBER,
  "IDINSTITUICAOCONTRATADO" NUMBER,
  "DATFINALVIGENCIA" NUMBER,
  "DATINICIOVIGENCIA" NUMBER,
  "DATASSINATURA" NUMBER,
  "VALEXECFINANC" NUMBER,
  "VALTOTALADITIVO" NUMBER
)
/
-----
-- DDL for Table F_EXECUCAOFINANCEIRA_AGG
-----

CREATE TABLE "F_EXECUCAOFINANCEIRA_AGG"
(
  "VAL_ADITIVOS" NUMBER,
  "VAL_EXEC_FINANC" NUMBER,
  "QTR_KEY" NUMBER,
  "ID_EMPREENDIMENTO" NUMBER
)
/
-----
-- DDL for Index D_ANDAMENTOFINANCEIRO_PK
-----

CREATE UNIQUE INDEX "D_ANDAMENTOFINANCEIRO_PK" ON
"D_ANDAMENTOFINANCEIRO" ("IDANDAMENTOFINANCEIRO")
/
-----
-- DDL for Index D_EMPREENDIMENTO_PK
-----

```

```

CREATE UNIQUE INDEX "D_EMPREENDIMENTO_PK" ON "D_EMPREENDIMENTO"
("IDEMPREENDIMENTO")
/
-----
-- DDL for Index D_INSTITUICAO_PK
-----

CREATE UNIQUE INDEX "D_INSTITUICAO_PK" ON "D_INSTITUICAO"
("IDINSTITUICAO")
/
-----
-- DDL for Index D_TEMPO_PK
-----

CREATE UNIQUE INDEX "D_TEMPO_PK" ON "D_TEMPO" ("DAY_KEY")
/
-----
-- DDL for Index D_TEMPO_QTR_GRAIN_PK
-----

CREATE UNIQUE INDEX "D_TEMPO_QTR_GRAIN_PK" ON "D_TEMPO_QTR_GRAIN"
("QTR_KEY")
/
-----
-- DDL for Index F_EXECUCAOFINANCEIRA_PK
-----

CREATE UNIQUE INDEX "F_EXECUCAOFINANCEIRA_PK" ON
"F_EXECUCAOFINANCEIRA" ("IDEXECUCAOFINANCEIRA")
/
-----
-- Constraints for Table D_ANDAMENTOFINANCEIRO
-----

ALTER TABLE "D_ANDAMENTOFINANCEIRO" ADD CONSTRAINT
"D_ANDAMENTOFINANCEIRO_PK" PRIMARY KEY ("IDANDAMENTOFINANCEIRO")
ENABLE
ALTER TABLE "D_ANDAMENTOFINANCEIRO" MODIFY ("IDANDAMENTOFINANCEIRO"
NOT NULL ENABLE)
/
-----
-- Constraints for Table D_EMPREENDIMENTO
-----

ALTER TABLE "D_EMPREENDIMENTO" ADD CONSTRAINT "D_EMPREENDIMENTO_PK"
PRIMARY KEY ("IDEMPREENDIMENTO") ENABLE
ALTER TABLE "D_EMPREENDIMENTO" MODIFY ("IDEMPREENDIMENTO" NOT NULL
ENABLE)
/
-----
-- Constraints for Table D_INSTITUICAO
-----

ALTER TABLE "D_INSTITUICAO" ADD CONSTRAINT "D_INSTITUICAO_PK"
PRIMARY KEY ("IDINSTITUICAO") ENABLE
ALTER TABLE "D_INSTITUICAO" MODIFY ("IDINSTITUICAO" NOT NULL ENABLE)
/
-----
-- Constraints for Table D_TEMPO
-----

```

```

ALTER TABLE "D_TEMPO" ADD CONSTRAINT "D_TEMPO_PK" PRIMARY KEY
("DAY_KEY") ENABLE
ALTER TABLE "D_TEMPO" MODIFY ("DAY_KEY" NOT NULL ENABLE)
/

-----
-- Constraints for Table D_TEMPO_QTR_GRAIN
-----

ALTER TABLE "D_TEMPO_QTR_GRAIN" ADD CONSTRAINT
"D_TEMPO_QTR_GRAIN_PK" PRIMARY KEY ("QTR_KEY") ENABLE
ALTER TABLE "D_TEMPO_QTR_GRAIN" MODIFY ("QTR_KEY" NOT NULL ENABLE)
ALTER TABLE "D_TEMPO_QTR_GRAIN" MODIFY ("PER_NAME_QTR" NOT NULL
ENABLE)
ALTER TABLE "D_TEMPO_QTR_GRAIN" MODIFY ("PER_NAME_HALF" NOT NULL
ENABLE)
ALTER TABLE "D_TEMPO_QTR_GRAIN" MODIFY ("CAL_YEAR" NOT NULL ENABLE)
ALTER TABLE "D_TEMPO_QTR_GRAIN" MODIFY ("CAL_QTR" NOT NULL ENABLE)
ALTER TABLE "D_TEMPO_QTR_GRAIN" MODIFY ("CAL_HALF" NOT NULL ENABLE)
/

-----
-- Constraints for Table F_EXECUCAOFINANCEIRA
-----

ALTER TABLE "F_EXECUCAOFINANCEIRA" ADD CONSTRAINT
"F_EXECUCAOFINANCEIRA_PK" PRIMARY KEY ("IDEXECUCAOFINANCEIRA") ENABLE
ALTER TABLE "F_EXECUCAOFINANCEIRA" MODIFY ("DATFINALVIGENCIA" NOT
NULL ENABLE)
ALTER TABLE "F_EXECUCAOFINANCEIRA" MODIFY
("IDINSTITUICAOCONTRATANTE" NOT NULL ENABLE)
ALTER TABLE "F_EXECUCAOFINANCEIRA" MODIFY ("IDANDAMENTOOFINANCEIRO"
NOT NULL ENABLE)
ALTER TABLE "F_EXECUCAOFINANCEIRA" MODIFY ("IDEMPREENDIMENTO" NOT
NULL ENABLE)
ALTER TABLE "F_EXECUCAOFINANCEIRA" MODIFY ("IDEXECUCAOFINANCEIRA"
NOT NULL ENABLE)
/

-----
-- Constraints for Table F_EXECUCAOFINANCEIRA_AGG
-----

ALTER TABLE "F_EXECUCAOFINANCEIRA_AGG" MODIFY ("ID_EMPREENDIMENTO"
NOT NULL ENABLE)
ALTER TABLE "F_EXECUCAOFINANCEIRA_AGG" MODIFY ("QTR_KEY" NOT NULL
ENABLE)
/

-----
-- Ref Constraints for Table F_EXECUCAOFINANCEIRA
-----

ALTER TABLE "F_EXECUCAOFINANCEIRA" ADD CONSTRAINT
"DW_EXECFINANC_INSTCONTRATANTE" FOREIGN KEY
("IDINSTITUICAOCONTRATANTE")
REFERENCES "D_INSTITUICAO" ("IDINSTITUICAO") ENABLE
ALTER TABLE "F_EXECUCAOFINANCEIRA" ADD CONSTRAINT
"DW_EXECUCAOFINANCEIRA_TEMPO2" FOREIGN KEY ("DATINICIOVIGENCIA")
REFERENCES "D_TEMPO" ("DAY_KEY") ENABLE
ALTER TABLE "F_EXECUCAOFINANCEIRA" ADD CONSTRAINT
"DW_EXECUCAOFINANCEIRA_TEMPO3" FOREIGN KEY ("DATASSINATURA")
REFERENCES "D_TEMPO" ("DAY_KEY") ENABLE
ALTER TABLE "F_EXECUCAOFINANCEIRA" ADD CONSTRAINT
"DW_EXECUCAOFINANCEIRA_TEMPO_FK" FOREIGN KEY ("DATFINALVIGENCIA")

```

```

REFERENCES "D_TEMPO" ("DAY_KEY") ENABLE
ALTER TABLE "F_EXECUCAOFINANCEIRA" ADD CONSTRAINT
"DW_EXECUCAOFINANC_ANDAME_FK2" FOREIGN KEY ("IDANDAMENTOFINANCEIRO")
REFERENCES "D_ANDAMENTOFINANCEIRO" ("IDANDAMENTOFINANCEIRO")
ENABLE
ALTER TABLE "F_EXECUCAOFINANCEIRA" ADD CONSTRAINT
"DW_EXECUCAOFINANC_EMPREE_FK1" FOREIGN KEY ("IDEMPREENDIMENTO")
REFERENCES "D_EMPREENDIMENTO" ("IDEMPREENDIMENTO") ENABLE
/

```

## DDL usada no PostgreSQL

```

-----
-- DDL usada no PostgreSQL
-----

CREATE TABLE copa.d_andamentofinanceiro
(
  idandamentofinanceiro integer NOT NULL,
  descandamentofinanceiro character varying(255),
  CONSTRAINT d_andamentofinanceiro_pkey PRIMARY KEY
(idandamentofinanceiro)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE copa.d_andamentofinanceiro
OWNER TO postgres;

CREATE TABLE copa.d_empreendimento
(
  idempreendimento integer NOT NULL,
  idcidade sede integer,
  descempreendimento character varying(4000),
  descobservacao character varying(4000),
  txtexplicativo investimento character varying(4000),
  desccidade sede character varying(255),
  idtema integer,
  desc tema character varying(255),
  CONSTRAINT d_empreendimento_pkey PRIMARY KEY (idempreendimento)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE copa.d_empreendimento
OWNER TO postgres;

CREATE TABLE copa.d_instituicao
(
  idinstituicao integer NOT NULL,
  idtipoinstituicao integer,
  nominstituicao character varying(255),
  desc tipoinstituicao character varying(255),
  esferatipoinstituicao character varying(255),
  CONSTRAINT d_instituicao_pkey PRIMARY KEY (idinstituicao)
)
WITH (
  OIDS=FALSE

```

```

);
ALTER TABLE copa.d_instituicao
  OWNER TO postgres;

CREATE TABLE copa.d_tempo
(
  calendar_date date,
  day_key integer NOT NULL,
  cal_half integer,
  cal_month integer,
  cal_qtr integer,
  cal_year integer,
  day_of_month integer,
  per_name_half character varying(255),
  per_name_month character varying(255),
  per_name_qtr character varying(255),
  CONSTRAINT d_tempo_pkey PRIMARY KEY (day_key)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE copa.d_tempo
  OWNER TO postgres;

CREATE TABLE copa.f_execucaofinanceira
(
  idexecucaofinanceira integer NOT NULL,
  idempreendimento integer NOT NULL,
  idandamentofinanceiro integer NOT NULL,
  idinstituicaocontratante integer NOT NULL,
  idinstituicaocontratado integer,
  datfinalvigencia integer NOT NULL,
  datiniciovigencia integer,
  datassinatura integer,
  valexecfinanc double precision,
  valtotaladitivo double precision,
  CONSTRAINT f_execucaofinanceira_pkey PRIMARY KEY
(idexecucaofinanceira),
  CONSTRAINT dw_execfinanc_instcontratante FOREIGN KEY
(idinstituicaocontratante)
  REFERENCES copa.d_instituicao (idinstituicao) MATCH FULL
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT dw_execucaofinanc_andame_fk2 FOREIGN KEY
(idandamentofinanceiro)
  REFERENCES copa.d_andamentofinanceiro (idandamentofinanceiro)
MATCH FULL
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT dw_execucaofinanc_empree_fk1 FOREIGN KEY
(idempreendimento)
  REFERENCES copa.d_empreendimento (idempreendimento) MATCH FULL
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT dw_execucaofinanceira_tempo2 FOREIGN KEY
(datiniciovigencia)
  REFERENCES copa.d_tempo (day_key) MATCH FULL
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT dw_execucaofinanceira_tempo3 FOREIGN KEY (datassinatura)
  REFERENCES copa.d_tempo (day_key) MATCH FULL
  ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT dw_execucaofinanceira_tempo_fk FOREIGN KEY
(datfinalvigencia)
  REFERENCES copa.d_tempo (day_key) MATCH FULL

```

```
        ON UPDATE NO ACTION ON DELETE NO ACTION
    )
    WITH (
        OIDS=FALSE
    );
ALTER TABLE copa.f_execucaofinanceira
    OWNER TO postgres;
```