

**Universidade Federal de Pernambuco  
Centro de Informática  
Graduação em Ciência da Computação**

---



**MDX+: Uma Linguagem de Consulta  
Multidimensional com Análise Horizontal, Vertical  
e Estatística**

---

**Trabalho de Graduação**

Thiago de Moraes Brayner

Recife, Dezembro de 2008

**Universidade Federal de Pernambuco  
Centro de Informática  
Graduação em Ciência da Computação**

---



**MDX+: Uma Linguagem de Consulta  
Multidimensional com Análise Horizontal, Vertical  
e Estatística**

---

**Trabalho de Graduação**

Thiago de Moraes Brayner

Trabalho apresentado ao Programa de  
Graduação em Ciência da Computação  
do Centro de Informática da  
Universidade Federal de Pernambuco  
como requisito parcial para a obtenção  
do grau de Bacharel em Ciência da  
Computação.

Orientadora: Dr<sup>a</sup> Valéria Cesário Times  
Co-Orientador: Paulo Caetano da Silva

Recife, Dezembro de 2008

# Agradecimentos

Agradeço a todos os meus colegas da turma 2004.2 que nesses quatro anos e meio de curso estiveram comigo a batalha da graduação.

Agradeço a minha família pelo apoio incondicional, em especial à minha mãe pela força e luta.

À Professora Valéria Times, que me orientou e incentivou no desenvolvimento deste trabalho.

A Paulo Caetano da Silva pela ajuda e contribuição neste trabalho.

# Resumo

No mundo moderno e globalizado em que vivemos, torna-se necessário a utilização de ferramentas que proporcionem aos gerentes e diretores executivos, uma melhor análise sobre os dados das empresas e grandes corporações. Ferramentas OLAP possibilitam diversas formas de análise dos dados, extraindo informações relevantes para a tomada de decisão estratégica. Porém, não contemplam certos tipos de análise amplamente utilizados, como análise por meio de índices, análise vertical, horizontal e percentil. Este trabalho propõe uma extensão da linguagem MDX, que possibilite esses tipos de análise.

**Palavras-chave:** Sistemas de Suporte à Decisão, MDX, análise por meio de índices, análise vertical, análise horizontal.

# Sumário

1.	Introdução .....	1
1.1.	Motivação.....	1
1.2.	Objetivos .....	2
1.3.	Estrutura do Documento .....	2
2.	Conceitos Básicos .....	4
2.1.	Sistemas de Suporte à Decisão.....	4
2.2.	Sistemas de Data Warehouse .....	5
2.3.	OLAP .....	6
2.4.	MDX.....	9
2.5.	Mondrian .....	10
2.6.	Conclusão .....	13
3.	Tipos de Análise.....	14
3.1.	Análise por meio de Índices.....	14
3.2.	Análise Vertical .....	15
3.3.	Análise Horizontal .....	16
3.4.	Análise Estatística.....	17
3.5.	Conclusão .....	19
4.	MDX+ .....	20
4.1.	Requisitos de MDX+ .....	20
4.2.	Arquitetura MDX+ .....	21
4.2.1.	Camada de Apresentação .....	22
4.2.2.	Camada de Processamento .....	23
4.2.3.	Camada de Armazenamento .....	24
4.3.	Processamento de consultas MDX+ .....	26
4.4.	Sintaxe .....	27
4.5.	Operadores de Criação de Fatos Calculados .....	28
4.5.1.	Operador CriarFatoCalculado.....	29
4.5.2.	Operador CriarFatoCalculadoRecurso.....	29
4.6.	Operadores UDF (Análise Vertical/Horizontal e Percentil).....	30

4.6.1. Análise Vertical .....	31
4.6.2. Análise Horizontal .....	32
4.5.3. Operador Percentil .....	34
5. Estudo de Caso .....	35
5.1 Data Mart Financeiro .....	35
5.2 Utilizando MDX+ com Data Mart Financeiro .....	36
5.2.1 Criação Índice RentabilidadeAtivo .....	36
5.2.2 Análise Vertical .....	37
5.2.3 Análise Horizontal .....	37
5.2.4 Análise Percentil .....	38
6. Conclusão .....	40
6.1 Principais Contribuições .....	40
7. Bibliografia .....	41

# Lista de Figuras

Figura 2.1: Exemplo de Cubo de dados.....	8
Figura 2.2. Resultado da consulta MDX exibida no Quadro 2.2.....	10
Figura 2.3. Arquitetura do Mondrian.....	12
Figura 3.1: Exemplo de Análise Vertical.....	15
Figura 3.2: Exemplo de Análise Horizontal.....	16
Figura 4.1. Arquitetura da Linguagem MDX+.....	20
Figura 4.2. Tela de criação de fatos calculados.....	21
Figura 4.3: Camada de processamento MDX+.....	23
Figura 4.4. Processo de execução de consultas utilizando operadores MDX+.....	25
Figura 4.5 Diagrama de Classes da Linguagem MDX+.....	26
Figura 4.6. Estrutura hierárquica das classes de renderização.....	27

## Lista de Quadros

Quadro 2.1 Exemplo de uma Consulta MDX.....	10
Quadro 3.1 Índice de participação de capitais de terceiros (PCT) .....	15
Quadro 4.1 Gramática EBNF da linguagem MDX+ .....	26
Quadro 4.2 Gramática EBNF da linguagem MDX+ .....	27
Quadro 4.3 Criação do fato calculado PrecoMedio. ....	29
Quadro 4.4 Criação do fato calculado x.....	29
Quadro 4.5 Exemplo de consulta MDX com operador AnaliseVertical.....	32
Quadro 4.6 Exemplo de consulta MDX com operador AnaliseHorizontal.....	33
Quadro 4.7 Exemplo de consulta MDX com operador AnalisePercentil .....	34

# 1. Introdução

Este capítulo exhibe as motivações para a realização do trabalho descrito neste documento, lista os principais objetivos das atividades de pesquisa realizadas e define a estrutura deste documento.

## 1.1. *Motivação*

No mundo globalizado e competitivo de hoje, cada vez mais se faz necessário o uso de ferramentas que possibilitem uma melhor análise sobre os dados das empresas e grandes corporações facilitando dessa forma, a tomada de decisão.

Nas mais variadas áreas do conhecimento, é comum a utilização de índices para a extração de informação para a realização de certas análises sobre os dados. Índices podem ser comparados com valores padrões para se obter prognósticos, e também, com resultados obtidos por outras organizações para se verificar tendências e fornecer subsídios para o processo de tomada de decisão.

Análises comparativas desses índices podem ser feitas utilizando conceitos estatísticos como o percentil, que possibilita a geração de uma distribuição estatística dos dados para permitir comparações mais precisas.

Duas outras técnicas comumente utilizadas para a tomada de decisões estratégicas são Análise Vertical e Análise Horizontal dos dados. A análise vertical demonstra a relação existente entre os valores de um conjunto com um dos seus valores. Já a análise horizontal permite acompanhar a evolução de uma determinada informação ao longo de um período de tempo, tomando a informação de um período de tempo como base.

Ferramentas OLAP (*Online Analytical Processing* – Processamento Analítico On Line) são bastante utilizadas por organizações e fornecem um método de

acessar, visualizar, e analisar dados corporativos com alta flexibilidade e desempenho. Porém, não foi encontrada uma ferramenta OLAP na literatura de banco de dados e no mercado que permita ao usuário, a criação de seus próprios índices e o uso destes três tipos de análise citados acima.

## **1.2. Objetivos**

Esse trabalho tem como objetivo a extensão de uma linguagem de consulta multidimensional, que possibilite ao usuário a criação de índices para que uma melhor análise dos dados das organizações possa ser realizada. A linguagem proposta visa permitir a construção de índices baseados em outros anteriormente definidos, possibilitando o reuso e fornecendo maior flexibilidade aos usuários da linguagem. Está também no escopo deste trabalho, a incorporação de operadores de análise vertical, análise horizontal, assim como a inclusão do operador estatístico percentil, na linguagem MDX estendida.

Para o desenvolvimento da linguagem e de tais operadores, foi utilizado o servidor OLAP Mondrian e seu recurso UDF(*User Defined Functions*) que permite a criação de novos operadores para a linguagem MDX e seu uso em consultas a dados multidimensionais.

## **1.3. Estrutura do Documento**

Este documento está estruturado da seguinte forma:

- Capítulo 2 – Conceitos Básicos: Esse capítulo discute os conceitos introdutórios e apresenta as tecnologias utilizadas para a realização desse trabalho. Este capítulo discorre sobre Sistemas de Suporte à Decisão, Sistemas de *Data Warehouse*, ferramentas OLAP, a linguagem de consulta a dados multidimensionais MDX, e sobre o Mondrian, que foi o servidor

OLAP escolhido para a implementação da linguagem de consulta estendida e dos operadores de análise propostos.

- Capítulo 3 – Tipos de Análise: Define e exemplifica os tipos de análise abordados por este trabalho, dando uma visão geral do seu funcionamento e da sua importância no processo de tomada de decisão.
- Capítulo 4 – Implementação da Linguagem: Nesse capítulo, é apresentada a arquitetura do processador da linguagem de consulta estendida assim como a descrição de sua sintaxe para criação de novos índices. São também apresentados os operadores de análise vertical, horizontal e estatístico, criados a partir do recurso UDF do Mondrian.
- Capítulo 5 – Estudo de Caso: Nesse capítulo, é apresentada uma aplicação prática dos operadores definidos nesse trabalho. Estes operadores estão inseridos no contexto de uma análise financeira de uma organização.
- Capítulo 6 – Conclusão: São apresentadas as considerações finais e os trabalhos futuros relacionados.

## **2. Conceitos Básicos**

Neste Capítulo, são discutidas tanto as tecnologias quanto os conceitos básicos considerados importantes para o desenvolvimento deste trabalho, tais como Sistemas de Suporte à Decisão, Sistemas de *Data Warehouse* e Ferramentas OLAP. Finalmente, as características principais da linguagem de consulta MDX e do servidor OLAP Mondrian são também listadas neste capítulo.

### **2.1. Sistemas de Suporte à Decisão**

As tradicionais aplicações *On-Line Transaction Processing* (OLTP), responsáveis pela manipulação de dados operacionais, utilizados na realização das tarefas cotidianas das empresas, não são capazes de suprir as necessidades dos usuários na realização de análises estratégicas. Desta forma, a partir da necessidade de gerentes e diretores executivos das empresas para extraírem informações relevantes para a tomada de decisão estratégica para suas organizações, surgiram os Sistemas de Suporte à Decisão (SSD).

SSD possibilitam o gerenciamento e a análise, de forma eficiente e consistente, de grandes bases de dados permitindo inferir informações e novos conhecimentos que auxiliam usuários a compreender o comportamento dos dados de negócio de uma organização. Essas informações possibilitam que executivos tomem decisões estratégicas no intuito de melhor conduzir seu negócio, tomando decisões com mais eficiência, prevendo tendências e indicando fatos que poderiam passar despercebidos por muito tempo. Assim, o principal objetivo dos Sistemas de Suporte à Decisão é fornecer informações relevantes ao processo de tomada de decisões estratégicas de uma organização.

SSD diferenciam-se dos sistemas de suporte operacional pelo fato de que, enquanto os modelos de dados dos sistemas para suporte à decisão são

orientados por assunto (modelo estrela ou flocos de neve [1], [2]), os modelos de dados dos sistemas operacionais são orientados por transações (modelo de entidade e relacionamento [3], [2]). Assim, para implementar um Sistema de Suporte à Decisão eficiente, faz-se necessário dividir a arquitetura de dados da organização em dois ambientes de banco de dados [2]: (1) um específico para dados operacionais: estes bancos por via de regra já existem e têm aplicação e muitas operações já pré-definidas e (2) o outro, para suporte à decisão – estes bancos normalmente possuem uma arquitetura de *Data Warehouse* e são construídos a partir dos dados existentes nas bases de dados operacionais da organização.

Muitas tecnologias podem ser empregadas para a implementação de sistemas de suporte à decisão, entre elas *Data Warehouse* e OLAP, que serão descritas a seguir.

## **2.2. *Sistemas de Data Warehouse***

As informações contidas nos tradicionais sistemas orientados à transação não suprem as necessidades de consultas dos gerentes das empresas que precisam gerar relatórios gerenciais acessando dados de longos períodos históricos, muitas vezes de vários anos. Esse tipo de informação deve ser armazenada em um outro tipo de base de dados, dedicado exclusivamente para esse tipo de tarefa.

Um *Data Warehouse* é uma coleção de dados orientada por assuntos, integrada, variante no tempo, e não volátil, que tem por objetivo dar suporte aos processos de tomada de decisão [4]. Ela armazena os dados extraídos do ambiente de produção da empresa, que foram selecionados e depurados, tendo sido otimizados para o processamento de consulta de apoio à decisão e não para o processamento de transações. Além disso, ela possibilita o acesso a

informações confiáveis, com rapidez e garantia de manutenção da qualidade dos dados.

Portanto, um DW é um grande banco de dados, atualizado com pouca frequência e que normalmente é construído a partir de dados provenientes de várias fontes de dados diferentes. Ele Funciona como um provedor de informações de uma organização, pois concentra todas as informações estratégicas e históricas, extraídas dos sistemas transacionais. Assim, sistemas de DW se baseiam em modelos multidimensionais, os quais fornecem uma grande ajuda para as áreas de negócio, apoiando e melhorando o processo de tomada de decisões.

### **2.3. OLAP**

OLAP é uma tecnologia de software que permite que analistas, gerentes e executivos recuperem os dados de uma forma rápida, consistente e com acesso interativo para uma grande variedade de possíveis visões multidimensionais da informação gerada pela empresa. Mais sucintamente, OLAP é um conjunto de funcionalidades que tem, como principal objetivo, facilitar a análise multidimensional [4].

DW e OLAP são tecnologias complementares, enquanto DW é responsável pela forma como os dados estão armazenados, sistemas OLAP são responsáveis por extrair a informação de forma rápida, permitindo aos usuários, a análise de inúmeros cenários, não importando como e onde esses dados estão fisicamente armazenados, independente de ser em um DW ou não.

Segundo [5], aplicações OLAP devem atender doze regras básicas, entre as quais está o conceito de visão multidimensional que se tornou a característica fundamental para o desenvolvimento desse tipo de aplicação. A visão multidimensional consiste na realização de consultas que fornecem dados a respeito de medidas de desempenho, decompostas por uma ou mais dimensões

dessas medidas. Além disto, os dados destas consultas podem também ser filtrados pela dimensão e/ou pelo valor da medida.

Estas visões multidimensionais são providas em função da estrutura dimensional do DW e podem ser compreendidas como eixos e pontos de um espaço multidimensional, no qual cada eixo pode ser visto como uma dimensão ou perspectiva (e.g. tempo, área geográfica, tipo de transação financeira) e os pontos, como um valor medido e correspondente à interseção desses eixos [6].

A representação da realidade, construída a partir de um modelo multidimensional, é obtida com base em um conjunto de conceitos. Os conceitos básicos modelados são fatos, medidas, dimensões e hierarquias. As definições de tais conceitos são dadas a seguir.

- Fato: é um dado de interesse para o processo de tomada de decisões, sendo importante que se tenham aspectos dinâmicos associados a ele, para que seja possível o registro de uma evolução com o passar do tempo. Um fato pode se referir, por exemplo, a transações de troca de estoque, empréstimos e operações bancárias. Portanto, a escolha de fatos pode ser baseada na periodicidade média de mudanças ou em algo de interesse específico para o analista.
- Medida: também chamada de variável ou métrica, é a propriedade numérica de um fato e descreve um dos seus aspectos quantitativos de interesses para a análise. A razão das medidas serem geralmente dados numéricos é que elas são utilizadas computacionalmente. Um exemplo de medida em um controle de vendas é a quantidade de unidades vendidas de certo produto.
- Dimensão: é uma propriedade de um fato com um domínio finito e descreve uma das suas perspectivas de análise. No exemplo de controle de vendas, dimensões típicas podem ser: produto, cliente e tempo. Recomenda-se que pelo menos uma das dimensões de um fato, represente a informação temporal.

- Nível de dimensão: é uma propriedade, com domínio finito, de uma dimensão. Por exemplo: um produto pode ser descrito por seu tipo, categoria e marca; um cliente, por sua cidade e país. Os relacionamentos entre níveis de dimensões são expressos por hierarquias.
- Hierarquia: é uma árvore direcionada e com raiz em uma dimensão, cujos nós são todos os níveis dessa dimensão, e cujos arcos simbolizam relações um-para-muitos entre os pares de níveis da dimensão. As hierarquias determinam como eventos primários podem ser agregados em secundários e selecionados significativamente para o processo de tomada de decisão.
- Membro: é um elemento que pertence à dimensão. Cada nível da hierarquia tem seu grupo de membros. Por exemplo, o nível *ano* da dimensão *tempo* pode ser representado pelos membros 2006, 2007 e 2008.
- Cubo: é a estrutura onde os dados estão armazenados e onde é processada a análise multidimensional dos dados através de consultas. Cada célula do cubo representa um valor ou uma medida. A Figura 2.1 exibe um exemplo de uma metáfora visual de um cubo com três dimensões.

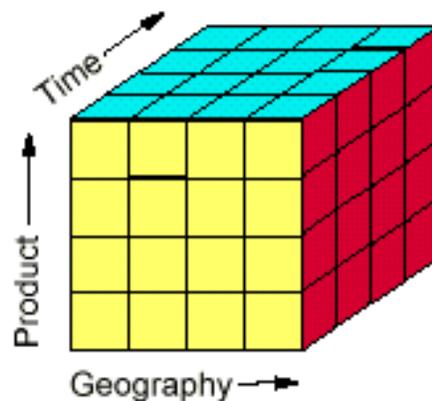


Figura 2.1: Exemplo de um cubo de dados

## 2.4. MDX

Sistemas OLAP utilizam linguagens de consulta para realizar a extração de informação em grandes volumes de dados, dessa forma ajudando as empresas na tomada de decisão.

MDX (*Multidimensional Expressions*) é uma linguagem de consulta criada pela *Microsoft Corporation* em 1998, para a utilização com o SQL Server OLAP Services como parte da especificação OLE DB/OLAP API. Ela permite que o usuário consulte cubos multidimensionais, retornando como resultado da consulta, conjuntos de células multidimensionais contendo dados desses cubos. Com a sintaxe MDX, torna-se muito fácil a aplicação dos operadores OLAP tradicionais, como é o caso das operações de agregação e desagregação (e.g. Roll-UP, Drill-Down e Group-By), e operações para a navegação do cubo de dados (e.g. Membros, Ancestor e Children), fornecendo visões configuráveis dos dados em diferentes ângulos e níveis de agregação. Ela é similar à linguagem padrão para consultas em sistemas relacionais SQL (*Structured Query Language*) em muitos aspectos, porém, não é uma extensão desta linguagem [7]. Ela permite a especificação de até 128 eixos em uma consulta. Porém é bastante incomum a utilização de mais de três dimensões. MDX é extensível, permitindo aos usuários, a criação e o registro de novas funções que manipulam dados multidimensionais.

MDX tornou-se a linguagem padrão para aplicações analíticas e é utilizada por inúmeras companhias, tanto aquelas que trabalham com OLAP no lado servidor como Applix, Microstrategy, SAS, SAP, Whitelight, NCR, quanto por companhias que trabalham com OLAP no lado cliente como Panorama, Proclarity, Cognos, Business Objects, Brio, Crystal. Por essas razões, ela é a linguagem de consulta para sistemas OLAP que foi escolhida para o desenvolvimento do trabalho descrito neste documento.

O Quadro 2.1 mostra um exemplo simples de uma consulta MDX: Esse exemplo realiza uma consulta ao cubo *Sales* especificado pelo comando FROM e

retorna uma tabela contendo a quantidade de unidades vendidas de produtos nas lojas do estado da Califórnia e no ano de 1997. A Figura 2.2 ilustra o resultado desta consulta MDX.

**Quadro 2.1. Exemplo de uma Consulta MDX.**

```
SELECT {[Time].[Year].[1997] } ON COLUMNS,  
{[Store].[Store Country].[USA].[CA].Children} ON ROWS  
FROM Sales  
WHERE ([Measures].[Unit Sales])
```

[Measures].[Unit Sales]	[Time].[1997]
[Store].[All Stores].[USA].[CA].[Alameda]	
[Store].[All Stores].[USA].[CA].[Beverly Hills]	21,333
[Store].[All Stores].[USA].[CA].[Los Angeles]	25,663
[Store].[All Stores].[USA].[CA].[San Diego]	25,635
[Store].[All Stores].[USA].[CA].[San Francisco]	2,117

**Figura 2.2. Resultado da consulta MDX exibida no Quadro 2.1**

Existe na literatura de banco de dados alguns trabalhos que propõem extensões de MDX em diversos tipos de aplicações. Entre elas, podemos citar a extensão de MDX com operadores espaciais, visando integrar funcionalidades OLAP com as de Sistemas de Integração Geográfico (SIG) [12]. Porém, não se tem conhecimento até o momento, sobre a existência de propostas que estendam MDX para permitir os tipos de análise aqui propostos.

## **2.5. Mondrian**

Mondrian é um servidor OLAP, de domínio público e de código aberto que foi desenvolvido pela *Pentaho Corporation* [8] e que processa consultas formuladas na linguagem MDX. Os dados processados pelo Mondrian são

mantidos em bancos de dados relacionais e seus resultados são apresentados em formato multidimensional. Ele foi desenvolvido na linguagem Java, sendo independente de plataforma, e podendo ser utilizado em diferentes sistemas operacionais.

Por ser de código aberto, o Mondrian permite a incorporação de novas funcionalidades ao servidor. Ele também disponibiliza um recurso para estender a linguagem MDX, denominado UDF (*User-Defined Functions*). A partir do uso de UDF, o usuário pode criar novos operadores que posteriormente podem ser usados em consultas MDX. Por esses motivos, o Mondrian foi escolhido como ferramenta OLAP para o desenvolvimento do trabalho proposto aqui.

A arquitetura do Mondrian contém quatro camadas: camada de apresentação, camada dimensional, camada estrela e a camada de armazenamento. Cada uma de suas camadas é detalhada a seguir.

- A camada de apresentação é a camada de interação com o usuário e nela, são encontradas as funcionalidades disponíveis e a forma como o usuário final pode interagir para submeter novas consultas. Os conjuntos de dados multidimensionais podem ser apresentados como *pivot tables*, ou como gráficos estáticos em diversos formatos (e.g. pizza, linha, barra). Ela também contém ferramentas avançadas de visualização, tais como mapas interativos e gráficos dinâmicos.
- A camada dimensional é responsável por fazer o *parsing*, e por validar e executar consultas MDX. A consulta é avaliada em diferentes fases. As dimensões contidas nos eixos são computadas primeiro e por último, os valores de cada célula são calculados.
- A terceira camada é a estrela, e é responsável por manter uma *cache* contendo um conjunto de valores de medida em memória. Ao receber uma

solicitação de consulta da camada dimensional, a *cache* é consultada, e caso ela tenha o valor solicitado, ele será retornado, caso contrario, a camada de armazenamento é invocada.

- A camada de armazenamento é formada pela base de dados relacional na qual são feitas as consultas OLAP.

A arquitetura do Mondrian é ilustrada pela Figura 2.3.

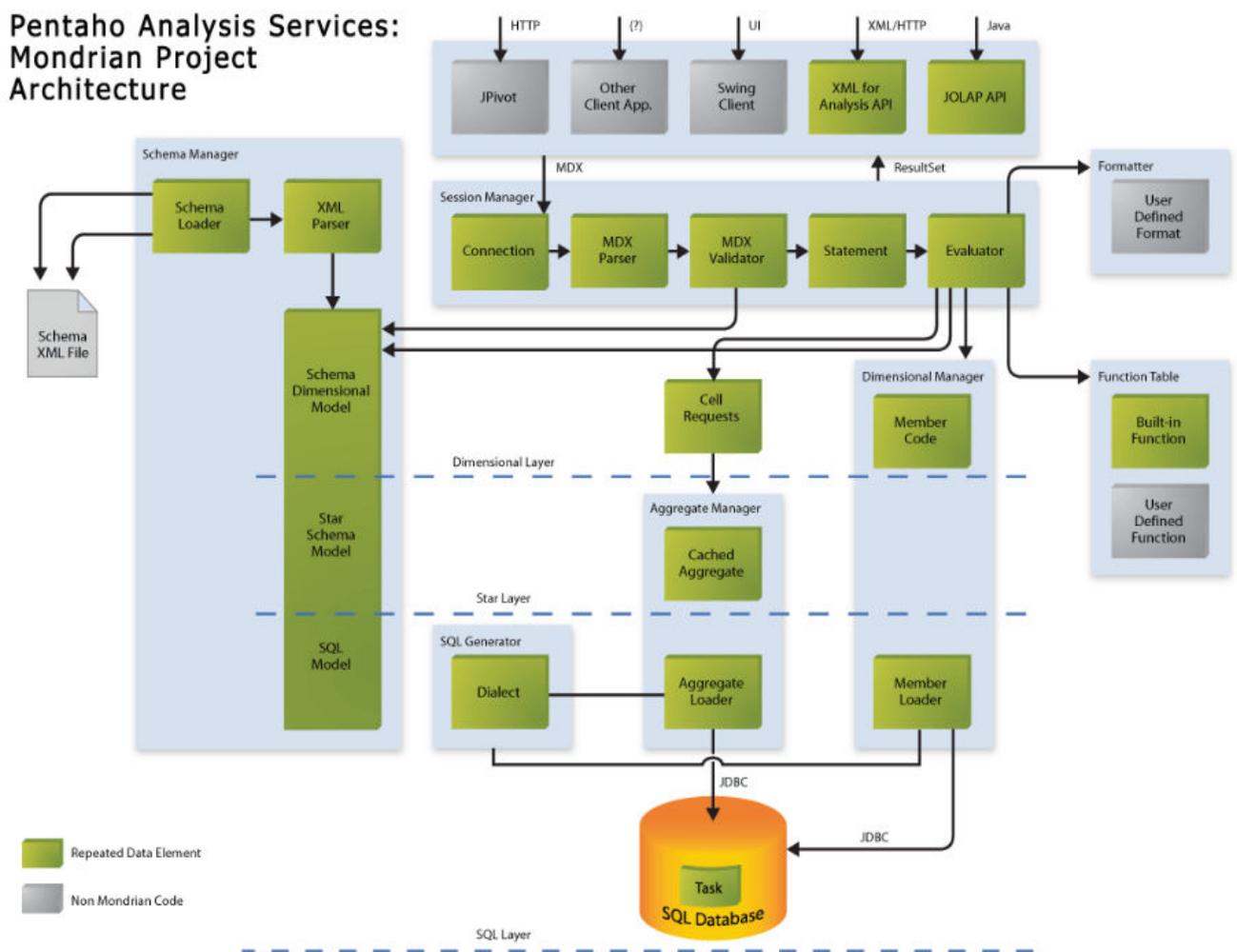


Figura 2.3. Arquitetura do Mondrian

## **2.6. Conclusão**

Este capítulo apresentou os principais conceitos introdutórios relativos a este trabalho. Desde conceitos básicos como SSD, passando por definições de DW, e OLAP. Em seguida foi apresentada a linguagem padrão de consulta a dados multidimensionais, chamada MDX e então, algumas de suas extensões foram discutidas, e por último, o servidor OLAP Mondrian utilizado para a implementação do trabalho foi descrito.

A seguir, serão discutidos os tipos de análise propostos por esse trabalho, apresentando uma visão geral da sua funcionalidade e enfatizando sua importância no processo de tomada de decisão.

### **3. Tipos de Análise**

Este capítulo discute a importância dos quatro tipos de análise propostos nesse trabalho como um complemento de consultas OLAP tradicionais. Inicialmente, é apresentada a análise por meio de índices, seguida pela discussão sobre análise vertical e horizontal e por último, o uso de algumas análises estatísticas para se obter comparações mais precisas é detalhado.

#### ***3.1. Análise por meio de Índices***

Índices são utilizados para análise de dados em uma variedade de contextos. Tais índices podem ser comparados com índices de períodos anteriores ou índices de outras companhias, dessa forma contribuindo para o processo de tomada de decisão.

Índices são relações que se estabelecem entre valores de uma mesma grandeza e suas aplicações se justificam quando se deseja ter uma análise mais precisa dos dados, visto que a observação de certas relações ou alguns percentuais é mais significativa do que a análise pura e simples dos dados disponibilizados nos relatórios [9].

Uma boa análise não é aquela realizada em cima de um grande número de índices. Melhores resultados podem ser obtidos a partir da escolha de um grupo de índices com boa representatividade, que permitam análises mais precisas. Como exemplo de índices, podemos destacar os índices econômicos e financeiros das empresas. Um exemplo desse tipo de índice é o que mede a participação de capitais de terceiros (PCT), que é calculado em relação ao Patrimônio Líquido, retratando a dependência da empresa em relação aos recursos externos. O índice PCT é apresentado no Quadro 3.1.

**Quadro 3.1. Índice de participação de capitais de terceiros (PCT)**

$$\text{PCT} = ((\text{PC} + \text{ELP}) / \text{PL}) \times 100$$

**Sendo:**

PC = Passivo Circulante

ELP = Exigível a Longo Prazo

PL = Patrimônio Líquido

### **3.2. Análise Vertical**

A Análise Vertical (AV) determina a participação de um item em relação ao grupo que ele faz parte. AV envolve a análise de elementos relativos sempre a um mesmo período e gerando os percentuais de cada item analisado. A Figura 3.1 demonstra um exemplo desse tipo de análise no contexto de gastos na área de saúde pública. É ilustrada a participação de cada item no total gasto, evidenciando a participação significativa que os gastos em assistência médica têm em relação aos demais

<b>Gastos do Governo Federal em programas de Saúde no ano de 2000</b>		
Vigilância Sanitária	R\$ 22.131.591	2.46%
Cont. Doenças Transmissíveis	R\$ 60.845.395	6.76%
Assistência Médica	R\$ 816.936.457	90.78%
Total	R\$ 899.913.443	100,00%

**Figura 3.1. Exemplo de Análise Vertical**

AV nos informa sobre a participação de uma determinada despesa em relação a um total, mas não nos diz se houve aumento ou diminuição em relação ao valor base considerado. Para responder esse tipo de pergunta, se faz

necessária uma análise complementar denominada de Análise Horizontal, que é definida na próxima seção.

### 3.3. Análise Horizontal

A Análise Horizontal (AH) é uma técnica que permite acompanhar a evolução de um elemento no decorrer de um determinado período de tempo. Essa evolução é evidenciada comparando-se o valor de um fato com o seu valor correspondente a um período anterior, que podemos chamar de período base. AH procura responder perguntas como:

- Quanto aumentou ou diminuiu do valor de cada item?
- Quais são os valores dos itens que mais aumentaram ou os que mais diminuiram em um dado período de tempo?

A Figura 3.2 exibe um exemplo desse tipo de análise no contexto de gastos na área de saúde pública. A partir de tal análise, podemos identificar: (1) um crescimento de quase 200% nos gastos em vigilância sanitária em 1999, em relação a 1998, e (2) tendências como a de queda nos investimentos no controle de doenças transmissíveis e (3) crescimento dos gastos em assistência médica, a partir do ano de 1999.

<b>Gastos do Governo Federal em programas de Saúde</b>						
	1998		1999		2000	
Vigilância Sanitária	9.570.883	100%	28.592.237	199%	22.131.591	-23%
Cont. Doenças Transmissíveis	82.871.010	100%	74.481.857	-10%	60.845.395	-18%
Assistência Médica	443.537.520	100%	642.100.381	45%	816.936.457	27%

**Figura 3.2. Exemplo de Análise Horizontal**

Os resultados obtidos por meio da análise horizontal devem ser interpretados com reserva, porque nem sempre os maiores valores percentuais de

aumento são os mais significativos [10]. É o caso do aumento de 199% em 1999, nos gastos com vigilância sanitária, que representa um acréscimo de R\$ 19.021.354, enquanto o de assistência médica, teve um crescimento bem menor, de apenas 45%, porém um acréscimo nos gastos de R\$ 198.562.861. Este último é um aumento bem mais significativo do que o com vigilância sanitária.

Por comparar valores de períodos diferentes, fatores externos podem influenciar nos resultados da análise. No exemplo da Figura 3.2, a inflação de cada ano tem que ser levada em consideração, antes de se fazer a análise. É necessário transformar todos os itens em valores associados a uma mesma data, para que possamos determinar a evolução real dos gastos.

Embora apresentem novas possibilidades de análise para ferramentas OLAP, estes dois tipos de análise apresentados acima não são suficientes para termos uma análise comparativa entre índices de empresas diferentes. Uma aplicação para este tipo de análise é a avaliação da situação financeira de uma dada empresa em relação à de outras empresas, que são do mesmo segmento, para saber, por exemplo, sua posição no mercado, ajudando os gerentes na tomada de decisão. A função *Rank* da linguagem MDX permite determinar a posição de determinado membro em relação a um conjunto de outros membros. Informa, por exemplo, a posição de classificação de uma empresa na venda de materiais esportivos na região nordeste do Brasil. Essa posição é calculada, comparando os valores absolutos de uma determinada medida desses membros. Porém, existem conceitos estatísticos que permite a extração de valores-padrão que possibilitem análises mais detalhadas, por exemplo, se uma empresa está entre os 10% das empresas que mais cresceram no ano passado. A análise estatística é descrita na seção seguinte.

### **3.4. Análise Estatística**

A análise estatística provê meios que possibilitam análises comparativas entre índices de períodos diferentes e índices pertencentes a outras instituições. Essa

análise é feita extraíndo-se índices-padrão que são usados como referência. Tomando como exemplo o índice pluviométrico de uma determinada região, podemos descobrir se esse índice se encontra acima ou abaixo da média dos 200 demais lugares pesquisados. Outros elementos estatísticos mais interessantes podem ser utilizados como a moda, a mediana e o decil. A Mediana é uma medida de localização que representa o centro da distribuição dos dados. Por exemplo, se entre os funcionários de uma determinada empresa, a mediana de salário for R\$ 2.400,00, aquele que receber remuneração acima disto, estará entre a metade dos funcionários mais bem remunerados da empresa. Porém, para análises mais detalhadas e precisas, faz-se necessário, a comparação com mais de um índice-padrão. O operador decil utiliza nove índices-padrão, o que permite uma análise mais detalhada dos dados.

Utilizando o decil, o universo das informações é dividido em fatias, cada uma com 10% dos elementos. Assim, através de decis, obtém-se uma idéia próxima de como se distribuem os elementos. No exemplo citado acima, os 200 índices pluviométricos comparados podem ser dispensados, e passa-se a trabalhar com apenas nove decis [11]. A Figura 3.3 ilustra a divisão do conjunto de informações em uma distribuição normal representado o decil.

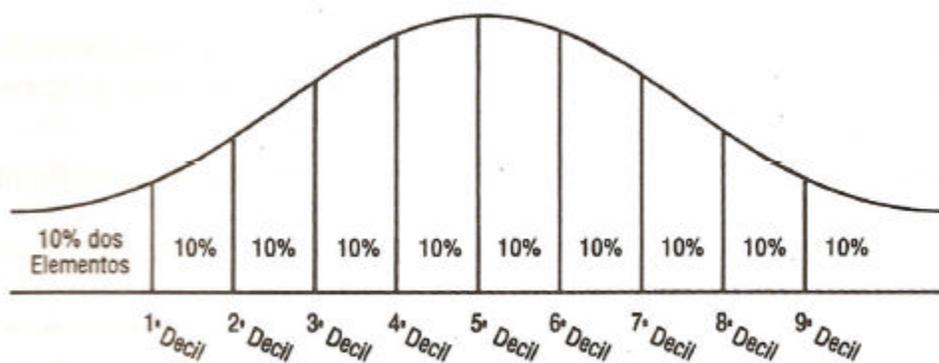


Figura 3.3 Distribuição do universo em 9 fatias

### **3.5. Conclusão**

Este capítulo exibiu uma discussão sobre análise vertical, análise horizontal e análise por meio de índices. Os tipos de análise vistos neste capítulo são de extrema importância quando objetivamos uma análise mais precisa dos dados. Portanto, a extensão de ferramentas OLAP com estas análises aumenta o universo de consultas de suporte à decisão realizadas pelos usuários. No próximo capítulo será apresentada uma extensão de MDX com operadores que permitem esses tipos de análise.

## 4. MDX+

Este capítulo apresenta a especificação da linguagem MDX+, uma extensão de MDX para prover recursos que permitam as análises discutidas no capítulo 3. Discute as características da linguagem proposta, sua arquitetura e também a sintaxe e a semântica dos operadores de análise.

### 4.1. *Requisitos de MDX+*

MDX+ é uma extensão da linguagem de consulta padrão para sistemas OLAP, e oferece funções que permitem ao usuário a possibilidade de análise vertical, horizontal e percentil dos dados multidimensionais e também a criação de índices de análise, e sua utilização por consultas MDX.

MDX+ foi implementada utilizando a linguagem de programação orientada a objetos Java, mantendo-a independente de plataforma por poder ser utilizada em diferentes sistemas operacionais. A linguagem foi desenvolvida estendendo o servidor OLAP Mondrian, que possibilita a incorporação de novas funcionalidades e criação de funções definidas pelo usuário. Ela provê aos usuários análise por meio de índices, através da criação de Fatos Calculados. Um Fato Calculado é definido neste trabalho como a representação de uma expressão aritmética aplicada a membros e medidas de um cubo de dados multidimensional.

Os fatos calculados são armazenados para permitir o uso em consultas MDX e também para possibilitar a criação de Fatos Calculados definidos a partir de outros anteriormente criados, sem a necessidade de reprocessar o cubo. MDX+ também adiciona operadores de análise Vertical, Horizontal e Percentil ao servidor Mondrian, a partir do seu recurso UDF.

## 4.2. Arquitetura MDX+

A arquitetura de MDX+ é dividida hierarquicamente em três camadas: camada de apresentação, camada de processamento e camada de armazenamento. São responsáveis respectivamente pela interface gráfica com o usuário, criação de fatos calculados / processamento de consultas utilizando fatos calculados e armazenamento dos índices definidos. Cada uma destas camadas é responsável por disponibilizar um determinado conjunto de serviços específicos à camada diretamente superior. A arquitetura em camadas foi escolhida, pois cada camada representa um subsistema independente dos outros, permitindo, por exemplo, a substituição da camada de apresentação por uma outra mais condizente com um determinado contexto, sem afetar o restante da hierarquia do sistema. A Figura 4.1 ilustra a arquitetura do processador da linguagem MDX+.

As próximas seções detalham cada uma das três camadas citadas anteriormente, descrevendo os serviços oferecidos por cada uma delas.

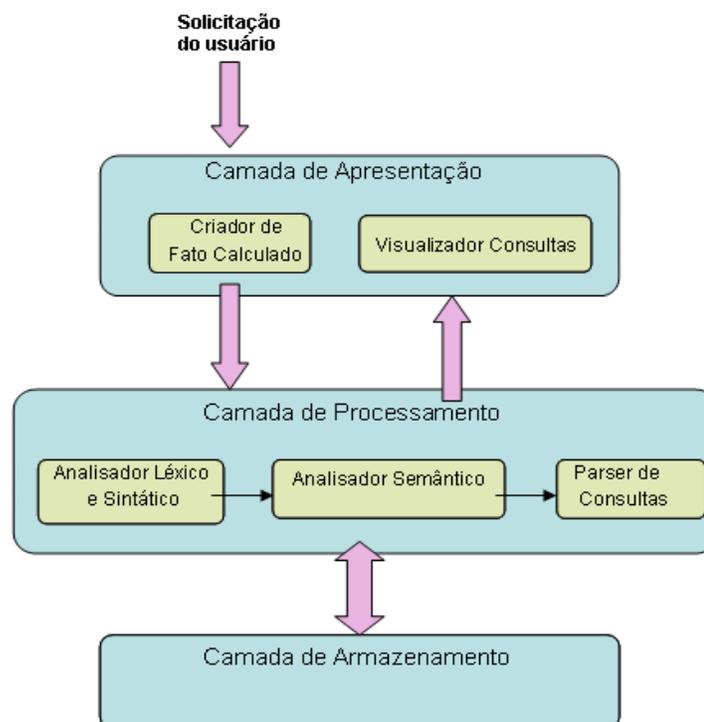


Figura 4.1. Arquitetura da Linguagem MDX+.

### **4.2.1. Camada de Apresentação**

A camada de apresentação é representada pela interface gráfica com usuário. Ela é constituída por dois componentes independentes. O primeiro deles é o Criador de Fato Calculado que permite a criação de novos fatos calculados de forma rápida e intuitiva. O segundo, é responsável por renderizar e permitir a visualização dos resultados obtidos a partir de consultas MDX, que utilizam os operadores de análise providos por MDX+. Ambos os componentes disponibilizam uma interface Web, e por esse motivo, podem ser acessados remotamente a partir de um *browser*, sem a necessidade de instalação de *softwares* auxiliares.

O componente Criador de Fato Calculado oferece ao usuário, uma interface amigável para a criação de novos fatos calculados visando seu posterior uso em consultas MDX. O componente disponibiliza uma lista de fatos calculados já criados e possibilita a visualização dos mesmos e seu uso na criação de novos fatos calculados.

O Visualizador de Resultados é responsável por disponibilizar ao usuário, os resultados gerados através da execução de consultas especificadas com operadores de análise MDX+. Estes resultados, são visualizados a partir de uma planilha de resultado específica para cada operador de análise. A Figura 4.3 ilustra o resultado de uma consulta utilizando o operador estatístico percentil, que neste exemplo divide o conjunto de dados em três fatias.

	[Measures].[Unit Sales]	
	16412.0	25323.0
[Store].[All Stores].[USA].[WA].[Tacoma]		35.257
[Store].[All Stores].[USA].[CA].[Beverly Hills]	21.333	
[Store].[All Stores].[USA].[CA].[Los Angeles]		25.663
[Store].[All Stores].[USA].[CA].[San Diego]		25.635
[Store].[All Stores].[USA].[CA].[San Francisco]	2.117	
[Store].[All Stores].[USA].[OR].[Portland]		26.079
[Store].[All Stores].[USA].[OR].[Salem]		41.580
[Store].[All Stores].[USA].[WA].[Bellingham]	2.237	
[Store].[All Stores].[USA].[WA].[Bremerton]		24.576
[Store].[All Stores].[USA].[WA].[Seattle]		25.011
[Store].[All Stores].[USA].[WA].[Spokane]		23.591
[Store].[All Stores].[USA].[WA].[Tacoma]		35.257
[Store].[All Stores].[USA].[WA].[Walla Walla]	2.203	
[Store].[All Stores].[USA].[WA].[Yakima]	11.491	

Figura 4.3. Resultado de consulta com o operador percentil.

#### 4.2.2. Camada de Processamento

A camada de Processamento é o núcleo do processador da linguagem estendida e é responsável por analisar e verificar a corretude da especificação de novos fatos calculados. É responsável também pelo processamento de consultas envolvendo os fatos calculados criados por MDX+. Esta camada é formada por três componentes, o Analisador Léxico e Sintático, o Analisador Semântico e o Parser de Consultas os quais são detalhados a seguir.

O Analisador Léxico e Sintático tem como principal responsabilidade, analisar a seqüência de caracteres dados como entrada. Ele verifica possíveis erros de sintaxe, classificando a string de entrada que representa a definição do novo índice, como pertencente ou não à linguagem. Um possível erro de sintaxe pode ser a utilização de um comando inexistente em sua gramática EBNF. O Analisador Semântico valida, por exemplo, o tipo dos argumentos passados como parâmetro.

A Figura 4.4 ilustra a seqüência de ações realizadas na camada de processamento quando o analisador léxico, sintático e semântico está em execução.

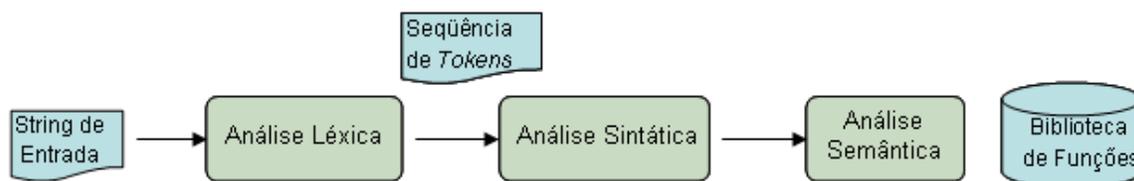


Figura 4.4: Camada de processamento MDX+

O Analisador Léxico transforma o texto dado como entrada em uma seqüência de *tokens*. Os *tokens* representam palavras (e.g. símbolos, identificadores, palavras-chave, pontuação) extraídas da string de entrada. Os *tokens* identificados são utilizados como entrada para o Analisador Sintático, que é responsável por avaliar sua corretude. O Analisador verifica a seqüência de *tokens* para determinar a estrutura das frases, verificando se um determinado identificador é sintaticamente válido, ou seja, se pertencem à especificação da EBNF de MDX+. Caso seja válido, o Analisador Semântico é invocado para verificar se o comando está de acordo com as restrições contextuais da linguagem MDX+. Por exemplo, se o tipo de um determinado valor passado como parâmetro está correto.

O Parser de Consultas é o módulo responsável pela transformação das consultas que utilizam fatos calculados. O resultado deste *parser* é uma consulta MDX válida, que é enviada para o servidor OLAP para ser processada.

#### 4.2.3. Camada de Armazenamento

A camada de armazenamento é responsável por gerenciar a persistência e a posterior recuperação dos fatos calculados criados. Os fatos são armazenados em arquivo e podem ser reutilizados por usuários diferentes do autor dos fatos.

Os módulos que compõem a arquitetura do processador de MDX+ e que foram descritos nas seções anteriores são representados no diagrama de classes ilustrado na Figura 4.5.

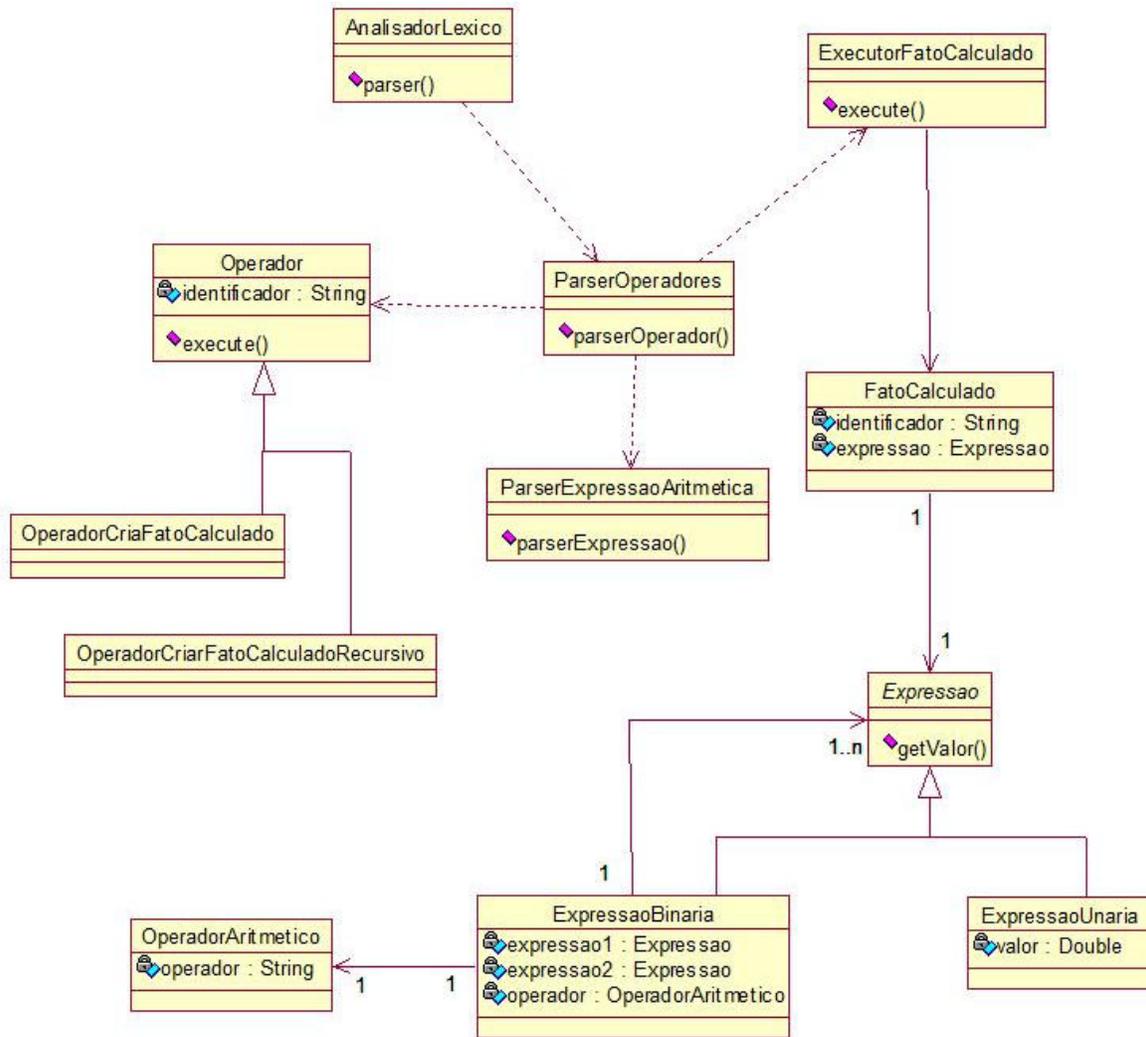


Figura 4.5 Diagrama de Classes da Linguagem MDX+

As classes *AnalisadorLexico*, *ParserOperador*, *ParserExpressao* e *ParserConsulta* constituem a camada de Processamento. O *AnalisadorLexico* realiza a tarefa de quebrar a *string* de entrada em *tokens*, para facilitar o processamento dos analisadores sintáticos. O *ParserOperador* é invocado para realizar a análise nos *tokens*. A avaliação da expressão aritmética é processada e

avaliada pela classe *ParserExpressão*. A classe *FatoCalculado* é constituída por uma identificador e uma objeto do tipo *Expressão*, que pode ser uma *ExpressaoUnaria* ou *ExpressaoBinaria*.

A estrutura hierárquica das classes responsáveis pela renderização das planilhas de resultado das consultas é apresentada pela Figura 4.6. Ela é formada por uma superclasse *GenericRender* e 3 subclasses específicas uma para cada operador de análise de MDX+.

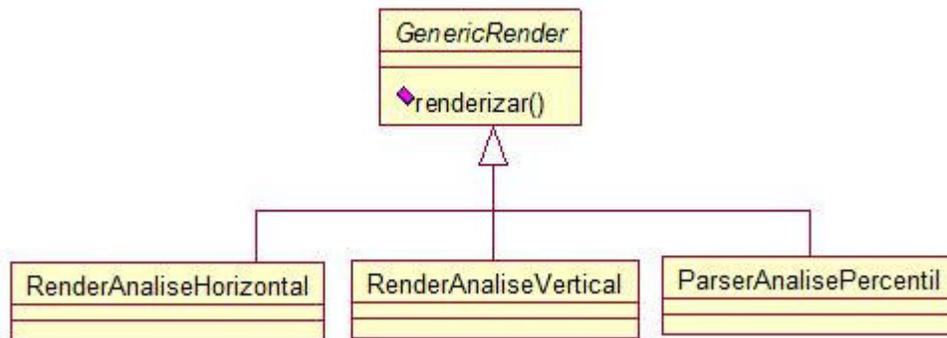


Figura 4.6. Estrutura hierárquica das classes de rederização.

### 4.3. Processamento de consultas MDX+

O processo de execução de consultas MDX+ é ilustrado pela Figura 4.7, e é descrito segundo a seqüência de passos listados a seguir.

1. Uma *string* representando a consulta MDX é submetida ao servidor Mondrian, a partir de sua interface gráfica de consultas.
2. A *string* de entrada é enviada à camada de processamento MDX+, que realiza análise léxica, dividindo a string em diversos tokens. Os tokens são enviados ao Analisador Sintático que verifica a existência de seqüências de tokens pertencentes à especificação da gramática EBNF de MDX+. Caso encontre, o Analisador Semântico processa tais seqüências com o objetivo de validá-las. Por

último, o *parser* de consultas é invocado para realizar a transformação em uma consulta MDX válida.

3. A consulta transformada é retornada ao processador de consultas do Mondrian, responsável por analisar, avaliar e executar a consulta MDX.

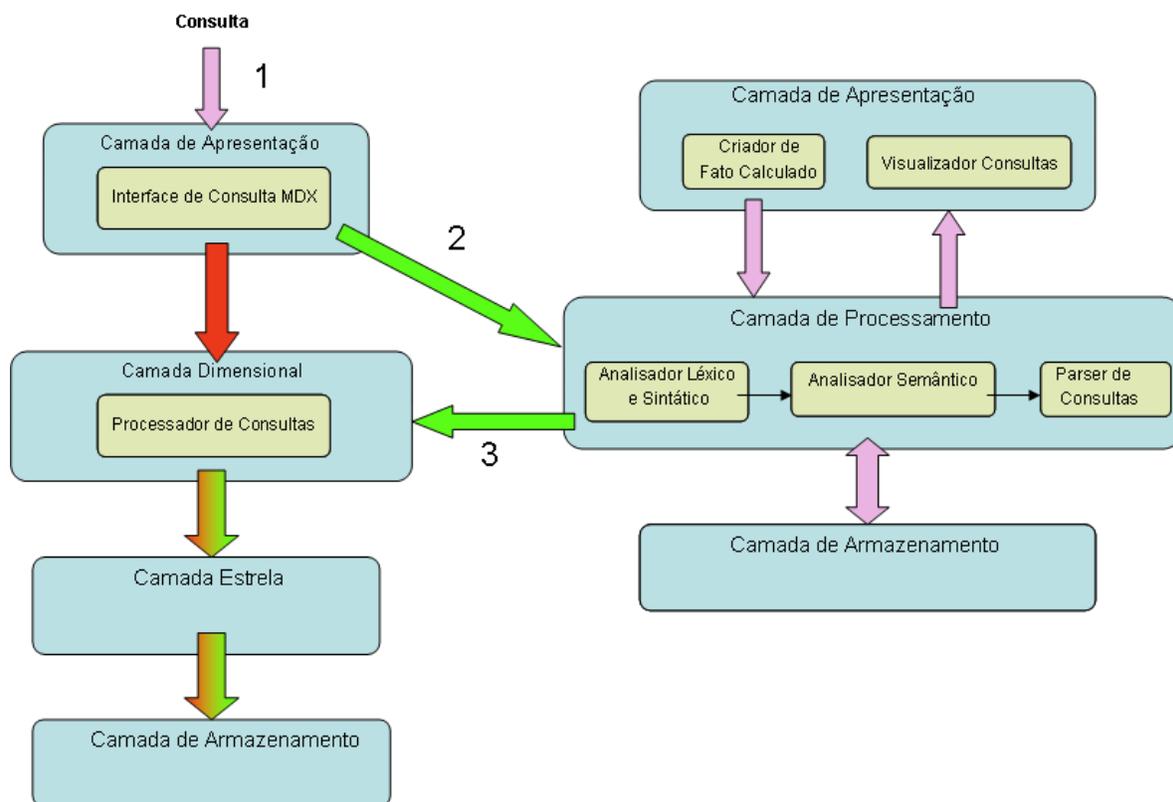


Figura 4.7 Processo de execução de consultas utilizando operadores MDX+

#### 4.4. Sintaxe

A linguagem MDX estendida oferece uma sintaxe simples e ao mesmo tempo rica, que permite a criação de fatos calculados representando expressões

aritméticas construídas a partir da utilização de seis operadores aritméticos: adição, subtração, multiplicação, divisão, potência e raiz enésima. Estes operadores foram implementados por serem considerados como o conjunto básico de operações aritméticas.

O Quadro 4.1 contém a especificação formal da sintaxe da linguagem descrita na EBNF (Extended Backus-Naur Form), uma metalinguagem utilizada para definir linguagem proposta de maneira formal e baseada em um conjunto de regras.

**Quadro 4.1 Gramática EBNF da linguagem MDX+**

Declaracao ::= Operador
Operador ::= Identificador "(" ListaParametros ")"
ListaParametros ::= [ ]
Parametro
Parametro "," ListaParametros
Parametro ::= Termo
Expressao
Expressao ::= ExpressaoUnaria
ExpressaoBinaria
ExpressaoUnaria ::= Termo
Identificador
ExpressaoBinaria ::= Expressao Operador Expressao
Operador ::= "+"   "-"   "*"   "/"   "^"   "raiz"
Termo ::= int
real
string

#### **4.5. Operadores de Criação de Fatos Calculados**

Nesta seção é apresentada a sintaxe, semântica e exemplos de uso dos operadores utilizados na criação de fatos calculados.

### 4.5.1. Operador CriarFatoCalculado

Índices podem ser criados a partir da função *CriarFatoCalculado*.

#### Sintaxe:

`CriarFatoCalculado(<Expressao>, <Identificador>)`

#### Semântica:

A função *CriarFatoCalculado* recebe dois parâmetros. O primeiro é o mais importante e representa a expressão aritmética do fato calculado. Ela especifica a forma como o índice é calculado. O segundo parâmetro é uma *string* que especifica o identificador do índice. MDX+ não permite dois fatos calculados com o mesmo identificador. Ao executar essa função um fato calculado é criado e armazenado, podendo ser usado por consultas MDX.

#### Exemplos:

O Quadro 4.2 apresenta um exemplo simples de criação de um fato calculado denominado *PrecoMedio*. Após a sua criação, o fato calculado *PrecoMedio* poderá ser utilizado na definição de outros fatos e também em consultas MDX

Quadro 4.2 Criação do fato calculado *PrecoMedio*.

```
CriarFatoCalculado ('([Medida].[Total Vendido] / [Medida].[Unidades Vendidas])',  
'PrecoMedio')
```

### 4.5.2. Operador CriarFatoCalculadoRecursivo

Alguns tipos de índice são por definição de natureza recursiva. Isto significa que eles podem ser definidos a partir da definição deles próprios. MDX+ permite a

criação dessa categoria de índices através do operador *CriarFatoCalculadoRecursivo*.

**Sintaxe:**

*CriarFatoCalculadoRecursivo* (<Expressao>, <ValorNumerico>, <ValorNumerico>, <String>)

**Semântica:**

A função *CriarFatoCalculadoRecursivo* recebe 4 parâmetros. O primeiro deles é a expressão aritmética do fato calculado. O segundo parâmetro é um valor numérico que representa o valor inicial do índice. O terceiro é um valor numérico que representa a quantidade de iterações que serão realizadas pelo índice sobre ele próprio. O quarto parâmetro é uma *string* que especifica o identificador do índice.

#### **4.6. Operadores UDF (Análise Vertical/Horizontal e Percentil)**

Para a criação de uma nova função MDX usando UDF, é necessário a criação de uma classe que implemente a interface *mondrian.spi.UserDefinedFunction*. Essa interface oferece sete métodos que especificam a sintaxe e a semântica da nova função e devem ser obrigatoriamente implementados. São eles:

- *getDescription()* - Essa função retorna uma descrição do novo operador MDX, e serve para facilitar o entendimento da funcionalidade do operador.
- *getName()* - Define o nome do operador. Esse método é utilizado pelo Mondrian para reconhecer as funções usadas na consulta MDX.
- *getReservedWords()* - Permite a definição de palavras reservadas e que não poderão ser utilizadas em nenhum outro operador definido com o recurso UDF.

- `getSyntax()` - Define como o Mondrian vai tratar essa nova UDF.
- `getParameterTypes()` - Esse método define quais são os tipos dos argumentos que devem ser passados como parâmetro para o novo operador. O Mondrian oferece vários tipos básicos já definidos como por exemplo: *MemberType*, *SetType*, *LevelType* e *NumericType*, representando respectivamente um membro, um conjunto de membros, um nível da hierarquia de uma dimensão e um valor numérico. Ele permite a criação de novos tipos dados, através da interface *mondrian.olap.type*.
- `getReturnType()` - Define o tipo de retorno do operador. Esse tipo pode ser qualquer um dos tipos já mencionados no item anterior ou um novo item definido pelo usuário.
- `execute()` - Método mais importante da UDF. É responsável pelo processamento do operador propriamente dito. Ele recebe como parâmetro, os argumentos da função e os processa, retornando assim, os resultados esperados.

Para criar os operadores de análise vertical e horizontal e o operador percentil, foi necessário utilizar a interface do `mondrian.spi.UserDefinedFunction` descrita acima. Nas próximas seções são detalhadas a sintaxe e semântica de cada um desses operadores.

#### **4.6.1. Análise Vertical**

O operador Análise Vertical permite uma análise comparativa entre um conjunto de membros em relação a um valor de referência.

**Sintaxe:**

AnaliseVertical(<Member>, <MemberSet>)

O operador recebe dois parâmetros: O primeiro é o membro de referência e será o valor relativo a esse membro que será usado no cálculo do percentual dos demais membros. O segundo parâmetro é um conjunto contendo os membros em que se quer saber a participação de cada um em relação ao membro de referência, o qual deve estar contido neste conjunto.

**Semântica:**

O operador retorna como resposta uma tabela contendo o valor absoluto e percentual de participação dos membros contidos no segundo parâmetro com relação ao primeiro parâmetro.

O Quadro 4.4 mostra um exemplo de consulta MDX utilizando o operador *AnaliseVertical*. Essa consulta retorna a porcentagem de participação de cada estado americano no total de unidades vendidas ao longo do ano de 2008.

**Quadro 4.4. Exemplo de consulta MDX com operador AnaliseVertical**

```
SELECT {[Time].[Year].[2008]} ON COLUMNS,  
{AnaliseVertical([Store].[StoreCountry].[USA],  
  {[Store].[Store Country].[USA].Members})} ON ROWS  
FROM Sales  
WHERE ([Measures].[Unit Sales])
```

#### **4.6.2. Análise Horizontal**

Esse operador permite a comparação de um determinado item ao seu valor em um período anterior.

**Sintaxe:**

AnaliseHorizontal(<Member>, <MemberSet>, NumericType)

O operador recebe três parâmetros: O primeiro (<Member>) é o período de referência e será o valor relativo a esse membro que utilizaremos no cálculo do percentual dos demais membros. O segundo parâmetro (<MemberSet>) é um conjunto contendo os membros em que se quer saber a evolução em relação ao valor de referência. Já o terceiro parâmetro é uma expressão que será usada para transformar os valores dos membros do segundo parâmetro em valores presentes de uma mesma data.

**Semântica:**

O operador retorna como resposta, uma tabela contendo o valor absoluto e percentual dos valores dos membros contidos no segundo parâmetro com relação ao primeiro parâmetro, levando em consideração a expressão passada no terceiro parâmetro.

O Quadro 4.5 demonstra um exemplo de consulta MDX utilizando o operador *AnaliseHorizontal*. A consulta retorna a evolução das vendas das lojas do estado da Califórnia de 2007 e 2008, aplicando o índice de correção 1.04 aos valores das vendas.

**Quadro 4.5. Exemplo de consulta MDX com operador AnaliseHorizontal**

```
SELECT {[Time].[Year].[2007], [Time].[Year].[2008]} ON COLUMNS,  
{AnaliseHorizontal([Time].[Year].[2007], {[Store].[Store Country].  
[USA].[CA].Children}, 1.04)} ON ROWS  
FROM Sales  
WHERE ([Measures].[Store Sales])
```

### 4.5.3. Operador Percentil

O operador percentil permite uma análise estatística para os dados multidimensionais.

#### Sintaxe:

AnalisePercentil(<MemberSet>, <MemberSet>, NumericType)

O operador recebe três parâmetros. O primeiro é um conjunto contendo os objetos para os quais se quer ter uma análise comparativa. O segundo parâmetro é um conjunto de membros utilizados para a criação dos percentis. Finalmente, o terceiro e último parâmetro é um valor numérico que informa a quantidade de partições em que o universo será dividido.

#### Semântica:

O operador retorna como resposta uma tabela representando a distribuição estatística construída a partir dos membros especificados no segundo parâmetro e da quantidade de partições descrita no terceiro parâmetro. Ele também informa a posição de cada membro do primeiro parâmetro no universo da distribuição.

O Quadro 4.6 mostra um exemplo do uso desse operador em uma consulta MDX. A consulta retorna a posição em relação à quantidade de unidades vendidas das cidades de Tacoma e Seattle em comparação com as outras cidades dos USA. A consulta gera uma distribuição do universo dividida em nove posições.

**Quadro 4.6. Exemplo de consulta MDX com operador AnalisePercentil**

```
SELECT {[Measures].[Unit Sales]} ON COLUMNS,  
{AnalisePercentil( {[Store].[Store Country].[USA].[WA].[Tacoma]}, [Store].[Store  
Country].[USA].[WA].[Seattle]}, {Descendants([Store].[Store Country].[USA], 2)},  
9)} ON ROWS  
FROM Sales
```

## 5. Estudo de Caso

Este capítulo apresenta o estudo de caso utilizado para a validação deste trabalho. Serão apresentados os resultados obtidos a partir dos experimentos realizados em um *data mart* financeiro.

### 5.1 Data Mart Financeiro

Para demonstrar os resultados desse trabalho foi escolhido um *Data Mart* com dados financeiros, por ser um contexto que utiliza uma grande variedade de índices para a análise da situação econômica e financeira das empresas. A Figura 6.1 ilustra o cubo de dados multidimensional deste data mart.

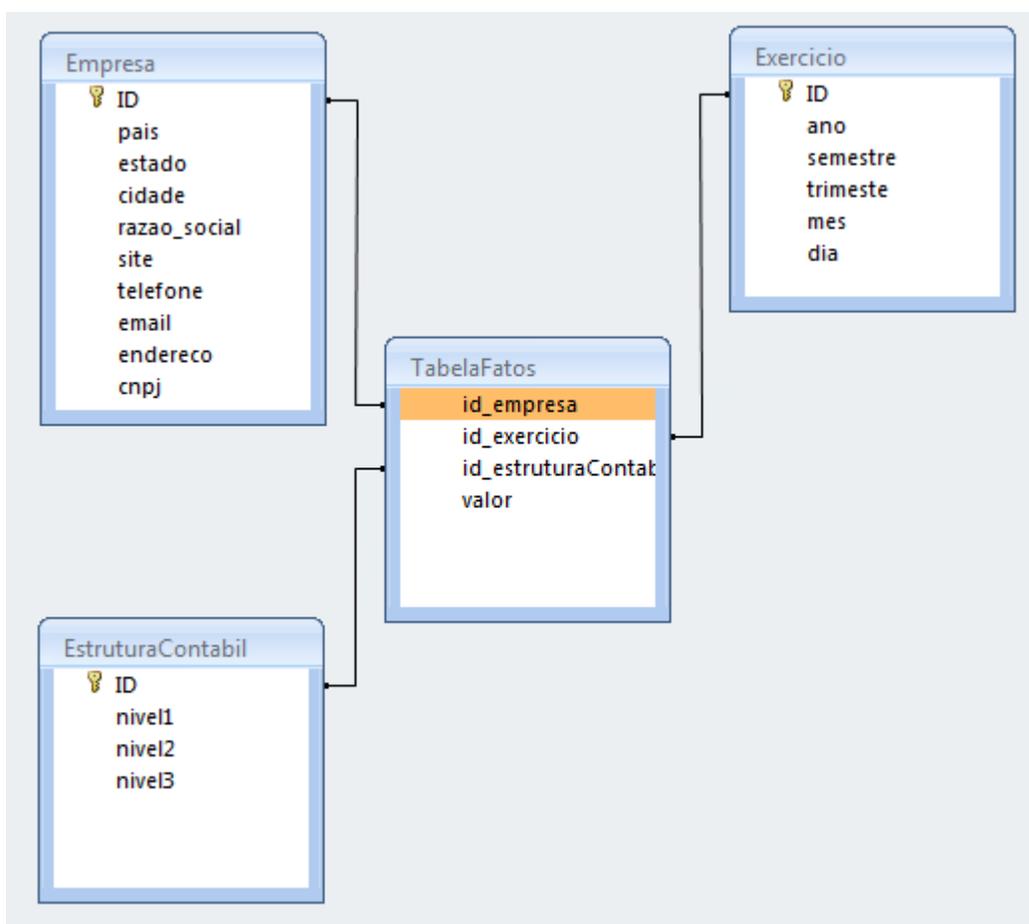


Figura 5.1 Cubo de dados multidimensional de um data mart financeiro.

## 5.2 Utilizando MDX+ com Data Mart Financeiro

Esta seção descreve a utilização dos operadores de análise vertical, horizontal e percentil e também a criação de índices a partir do uso de MDX+.

### 5.2.1 Criação Índice RentabilidadeAtivo

A rentabilidade do ativo é um índice que informa quanto a empresa obtém de lucro com relação ao investimento total. Este índice é calculado dividindo o lucro líquido pelo ativo da empresa.

O Quadro 5.1 apresenta o código de criação do índice *RentabilidadeAtivo*.

**Quadro 5.1 Código de criação do índice RentabilidadeAtivo.**

```
CriarFatoCalculado('([EstruturaContabil].[Lucro Liquido] / [EstruturaContabil].[Ativo])',  
'RentabilidadeAtivo')
```

**Quadro 5.2 Consulta MDX utilizando o índice RentabilidadeAtivo.**

```
SELECT {[Tempo].[ano].[2007]} ON COLUMNS,  
{DESCENDANTS ( [Empresa].[Brasil].[São Paulo], [razao social])} ON ROWS  
FROM Cubo_Financeiro  
WHERE (RentabilidadeAtivo)
```

[Measures].[RentabilidadeAtivo]	[Tempo].[All Tempos].[2007]
[Empresa].[All Empresas].[Brasil].[São Paulo].[Campinas].[Empresa5]	0 . 771
[Empresa].[All Empresas].[Brasil].[São Paulo].[Santos].[Empresa6]	0 . 641
[Empresa].[All Empresas].[Brasil].[São Paulo].[São Paulo].[Empresa4]	0 . 997
[Empresa].[All Empresas].[Brasil].[São Paulo].[São Paulo].[Empresa7]	0 . 646

**Figura 5.2 Resultado da consulta do Quadro 5.2.**

## 5.2.2 Análise Vertical

O Quadro 5.3 e a Figura 5.3 mostram a utilização de análise vertical para saber a porcentagem de participação dos itens do ativo operacional no ano de 2008.

Quadro5.3 Consulta MDX utilizando operador AnaliseVertical

```
SELECT {[Tempo].[ano].[2008]} ON COLUMNS,  
{AnaliseVertical([EstruturaContabil].[Ativo].[Ativo Operacional], {[EstruturaContabil].  
[Ativo].[Ativo Operacional].Children})} ON ROWS  
FROM Financeiro
```

	[Tempo].[All Tempos].[2008]	
	Valor Absoluto	AV (%)
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Operacional]	47.420	100.00
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Operacional].[Duplicadas a Receber]	18.300	38.59
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Operacional].[Estoques]	17.700	37.33
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Operacional].[Materia Prima]	2.970	6.26
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Operacional].[Produtos Acabdos]	3.560	7.51
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Operacional].[Produtos em Processo]	4.890	10.31

Figura 5.3 Resultado da consulta do Quadro 5.3

## 5.2.3 Análise Horizontal

O operador *AnaliseHorizontal* utilizado na consulta MDX do Quadro 5.4 descreve a evolução dos ativos financeiro, operacional e permanente ao longo dos anos de 2006 a 2008. A Figura 5.4 descreve o resultado desta consulta.

Quadro5.4 Consulta MDX utilizando operador AnaliseHorizontal

```
SELECT {[Tempo].[ano].Members} ON COLUMNS,
```

```
{AnaliseHorizontal([Tempo].[ano].[2006], {[EstruturaContabil].[Ativo].Children}, 1)}
ON ROWS
FROM Financeiro
WHERE ([Empresa].[Brasil].[Pernambuco].[Jaboatão].[Empresa3])
```

[Empresa].[All Empresas].[Brasil].[Pernambuco].[Jaboatão].[Empresa3]	[Tempo].[All Tempos].[2006]		[Tempo].[All Tempos].[2007]		[Tempo].[All Tempos].[2008]	
	Valor Absoluto	AH (%)	Valor Absoluto	AH (%)	Valor Absoluto	AH (%)
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Financeiro]	7.420	100.00	5.900	-125.76	8.000	107.82
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Operacional]	3.000	100.00	6.110	203.67	3.500	116.67
[EstruturaContabil].[All EstruturaContabils].[Ativo].[Ativo Permanente]	8.900	100.00	3.800	-234.21	3.540	-251.41

Figura 5.4 Resultado da consulta do Quadro 5.4

### 5.2.4 Análise Percentil

O Quadro 5.5 ilustra um exemplo de uma consulta MDX utilizando o operador AnalisePercentil e o índice RentabilidadeAtivo. A Figura 5.5 demonstra o resultado desta consulta.

Quadro 5.5 Consulta MDX utilizando operador AnaliseHorizontal

```
SELECT {RentabilidadeAtivo} ON COLUMNS,
{AnalisePercentil({[Empresa].[Brasil].[Pernambuco].[Recife].[Empresa3]},
Descendants([Empresa].[Brasil], [razao social]), 3)} ON ROWS
FROM Financeiro
```

	[Measures].[Rentabilidade.Ativo]	
	0,44	0,81
[Empresa].[All Empresas].[Brasil].[Pernambuco].[Recife].[Empresa3]	0,40	
[Empresa].[All Empresas].[Brasil].[Minas Gerais].[Belo Horizonte].[Empresa11]		0,89
[Empresa].[All Empresas].[Brasil].[Minas Gerais].[Belo Horizonte].[Empresa12]	0,24	
[Empresa].[All Empresas].[Brasil].[Pernambuco].[Recife].[Empresa1]	0,44	
[Empresa].[All Empresas].[Brasil].[Pernambuco].[Recife].[Empresa2]	0,43	
[Empresa].[All Empresas].[Brasil].[Pernambuco].[Recife].[Empresa3]	0,40	
[Empresa].[All Empresas].[Brasil].[Rio de Janeiro].[Rio de Janeiro].[Empresa10]	0,45	
[Empresa].[All Empresas].[Brasil].[Rio de Janeiro].[Rio de Janeiro].[Empresa8]		0,85
[Empresa].[All Empresas].[Brasil].[Rio de Janeiro].[Rio de Janeiro].[Empresa9]		3,32
[Empresa].[All Empresas].[Brasil].[São Paulo].[São Paulo].[Empresa4]		0,94
[Empresa].[All Empresas].[Brasil].[São Paulo].[São Paulo].[Empresa5]	0,77	
[Empresa].[All Empresas].[Brasil].[São Paulo].[São Paulo].[Empresa6]	0,64	
[Empresa].[All Empresas].[Brasil].[São Paulo].[São Paulo].[Empresa7]	0,65	

Figura 5.5 Resultado da consulta do Quadro 5.5

## **6. Conclusão**

Este trabalho propôs a criação de uma extensão para a linguagem padrão de consulta a dados multidimensional MDX, que possibilite ao usuário a criação de índices e análises vertical, horizontal e percentil dos dados.

A implementação da linguagem MDX+ envolveu a construção de um compilador capaz de realizar análises léxica, sintática e semântica dos comandos da linguagem, assim como o módulo responsável por renderizar os resultados das consultas MDX envolvendo operadores de análise horizontal, vertical e percentil.

### **6.1 Principais Contribuições**

A principal contribuição deste trabalho foi a especificação da linguagem MDX+. Ela permite uma forma simples de criação de índices e seu armazenamento para o posterior uso em consultas analíticas e também a construção de novos operadores de análise utilizando o recurso UDF do Servidor OLAP Mondrian, desta forma, permitindo aos usuários uma melhor análise sobre os dados, contribuindo para o processo de tomada de decisão.

## 7. Bibliografia

[1] Inmon, W. Building the DataWarehouse, 4th edn. John Wiley and Sons (2005)

[2] Kimball, R., Ross, M.: The DataWarehouse Toolkit. John Wiley and Sons (2002).

[3] R. Elmasri e S. B. Navathe. Fundamentals of Database Systems. Addison-Wesley, Hardcover, 3<sup>rd</sup> edition, 1999.

[4] INMON, W.H, WELCH, J. D., GLASSEY, K. L. Gerenciando Data Warehouse. São Paulo: Makron Books, 1999. 375p.

[5] Codd, E.: Providing OLAP (on-line analytical processing) to user-analysts: an IT mandate. Tech. rep., E.F. Codd and Associates (1993)

[6] Fidalgo. R. N. JDCI: Uma API Java para disponibilização e integração de serviços OLAP, Dissertação de Mestrado, Cin/UFPE, 2000.

[7]. MSDN. Microsoft Developer Network: <http://msdn.microsoft.com/pt-br/library/ms144884.aspx>

[8] Pentaho Open Source Business Intelligence: <http://www.pentaho.com/>

[9] ZOLET, J. P.; LOZECKYI, J. - *Concessão de Crédito para Micro e Pequena Empresa*

[10] UP-TO-DATE – N°294 – Como Elaborar e analisar uma análise vertical e horizontal das demonstrações financeiras.

[11] Matarazzo, D. C. (2003), *Análise Financeira de Balanços*, Editora. Atlas, São Paulo

[12] Worboys, M. e Duckham, M. (2004). "GIS: A Computing Perspective", 2nd Edition. CRC Press

## Assinaturas

---

Orientadora: Dr<sup>a</sup> Valéria Cesário Times

---

Co-Orientador: Paulo Caetano da Silva

---

Aluno: Thiago de Moraes Brayner