

# Tutorial MDX

autor: Enric Biosca  
[www.enricbiosca.es](http://www.enricbiosca.es)

Distruído bajo licencia Creative Commons

El objetivo de este breve tutorial es explicar los conceptos básicos del lenguaje MDX así como servir de introducción para que el lector puede dar sus primeros pasos con un motor que utilice MDX. Se utilizará en concreto la especificación de MDX para el servidor OLAP open source de la suite de BI Pentaho.

## ÍNDICE

### **0.- Breve introducción al análisis multidimensional y cubos OLAP**

#### **1.- ¿Que es MDX?**

#### **2.- Principales aspectos de la sintaxis de MDX**

#### **3.- Dimensiones de tiempo en MDX: funciones especiales**

#### **4.- Miembros calculados y formulas complejas.**

#### **5.- Bibliografía. Para saber más**

### **0.- Breve introducción al análisis multidimensional y cubos OLAP**

Antes de empezar a hablar de MDX debemos entender que significa el análisis multidimensional y los cubos OLAP pues el lenguaje MDX se utiliza para realizar consultas sobre estos modelos de datos.

El análisis multidimensional consiste en organizar la información y los datos de una organización de forma que los elementos medibles o métricas del negocio (también llamados hechos) puedan ser cualificados utilizando las diferentes "dimensiones" o vistas del negocio.

El objetivo de este modelo es permitir analizar una métrica cruzándola con diferentes puntos de vista del negocio. Pero entendámoslo mejor con un ejemplo:

Pongamos que tenemos un supermercado con una tarjeta de fidelización que nos permita saber quién compra que y cuando.

Un análisis u organización multidimensional de los datos de ventas de un supermercado nos podría permitir responder a preguntas como:

Que articulo es el más vendido este mes.

Que tienda ha vendido más artículos del tipo X

Que edad tienen los clientes que más gastan en mi supermercado (de media)

Veamos el paralelismo entre estas consultas y el modelo multidimensional.

¿Que articulo es el más vendido este mes?

En esta pregunta queremos analizar el hecho artículos vendidos y lo queremos analizar desde 2 dimensiones distintas: tiempo y familia de productos

¿Que supermercado ha vendido más artículos del tipo X?

El hecho a contar aquí es el mismo ( cantidad de artículos) pero las dimensiones son aquí las diferentes tiendas y la familia de productos

¿Que edad tienen los clientes que más gastan en mi supermercado ( de media)?  
En este caso el hecho es el importe de las ventas y la dimensión son los clientes, o bien la edad de los clientes, o bien su rango de edad. Lo veremos más adelante.

## 1.- Que es MDX

MDX es un acrónimo de Multidimensional Query eXpression. Este lenguaje fue creado en 1997 por Microsoft. No es un lenguaje estándar sin embargo diferentes fabricantes de herramientas OLAP(entre ellos Mondrian) lo han adoptado como estándar de hecho.

El MDX es en los sistemas OLAP el equivalente al SQL en los sistemas gestores de bases de datos relacionales. Eso significa que es el lenguaje a través del cual podemos explotar la información que reside en los motores OLAP y satisfacer las consultas analíticas de las que hemos hablado en el capítulo introductorio.

Es un lenguaje de consulta más cercano al lenguaje natural que el sql sin embargo tiene funciones y formulas que lo hacen muy potente para el análisis de datos.

La principal diferencia del mundo OLAP respecto al mundo relacional radica en que las estructuras dimensionales están jerarquizadas y se representan en forma de árbol y por lo tanto existen relaciones entre los diferentes miembros de las dimensiones.

Este hecho hace que el lenguaje MDX disponga de funciones y atributos especiales que permiten referenciar los diferentes elementos de las dimensiones con expresiones como MIEMBROS-HIJO , MIEMBROS-PRIMO, MIEMBROS-PADRE,etc... haciendo una analogía con los árboles genealógicos.

Estas estructuras jerarquizadas son especialmente útiles para poder visualizar los datos de forma comparada a nivel temporal pero de las dimensiones temporales y sus particularidades ya hablaremos en otro capítulo.

## 2.- Principales aspectos de la sintaxis de MDX

El lenguaje MDX permite realizar consultas con una sintaxis parecida al lenguaje natural. Para ejemplificar las consultas y la sintaxis trabajaremos con un ejemplo de un cubo.

Imaginamos un cubo de ventas con las siguientes dimensiones:

Dimensión temporal de las ventas con niveles de año y mes.

Dimensión productos vendidos con niveles de familia de productos y productos.

Dimensión de medidas con importe de las ventas y unidades vendidas

### 2.1.- Mi primera consulta

Si queremos obtener por ejemplo el importe de las ventas para el año 2008 para la familia de productos lácteos la consulta seria la siguiente:

```
SELECT
  { [Medidas].[importe ventas]}
on columns,
  { [Tiempo].[2008] }
on rows
FROM [cubo ventas]
WHERE ([Familia].[lacteos])
```

### 2.2.- Analogía con SQL

Analizando la consulta de ejemplo podemos establecer una analogía con el sql para explicar mejor la estructura del lenguaje. En primer lugar tenemos la estructura idéntica general de la consulta con las cláusulas **SELECT ... FROM ... WHERE** que en MDX se usa idénticamente al sql especificando en el select un conjunto de elementos que queremos visualizar, en el from indicamos el cubo del que queremos sacar la información y en el where las condiciones de filtrado.

Una de las particularidades del MDX es que la información la devuelve en forma tabular normalmente y debe ser procesada. Si os fijáis en la cláusula select se usa la

clausula on columns. Esta cláusula sirve para separar los elementos que queremos visualizar en las filas de los que queremos en las columnas. En la consulta de ejemplo veremos el importe en las columnas y el tiempo en las filas. Los { } son para permitir listas de elementos en las selecciones y los [] encapsulan elementos de las dimensiones y niveles.

### 2.3.- Principales funciones, opciones y elementos MDX

A nivel de funcionalidades y potencia a la hora de realizar consultas el MDX es potente como el SQL aunque su objetivo está orientado a temas de comparaciones, relaciones jerárquicas entre elementos, etc...

Una de las funcionalidades que distinguen al MDX es el acceder a los elementos utilizando estructura de árbol. Así para un Determinado nivel de una dimensión tenemos comandos como:

[Familia].[lácteos].CurrentMember : Permite acceder al miembro actual

[Familia].[lácteos].Children : Permite acceder a los hijos de la familia de los

lácteos

[Familia].[lácteos].prevMember: Permite acceder al miembro anterior de la dimensión

Por otro lado existen diferentes funciones que permiten realizar cálculos y complementar las consultas como por ejemplo:

CrossJoin(conjunto\_a, conjunto\_b): Obtiene el producto cartesiano de dos conjuntos de datos.

BottomCount(conjunto\_datos, N): Obtener un número determinado de elementos de un conjunto, empezando por abajo, opcionalmente ordenado.

BottomSum(conjunto\_datos, N, S): Obtener de un conjunto ordenado los N elementos cuyo total es como mínimo el especificado( S).

Except(conjunto\_a, conjunto\_b): Obtener la diferencia entre dos conjuntos.

Finalmente comentar que en MDX también están implementadas muchas funciones matemáticas y estadísticas que permiten enriquecer los análisis. Encontramos entre muchas otras funciones como:

AVG

COUNT

VARIANCE

VARIANCEn

y todas las funciones trigonométricas (Seno, Coseno, tangente, etc...)

## 3.- Dimensiones de tiempo en MDX: funciones especiales

Como ya hemos comentado a lo largo de este tutorial una de las funcionalidades destacadas del lenguaje MDX son las funciones especiales de tratamiento del tiempo. En el análisis multidimensional es muy frecuente comparar los valores de las métricas con los valores de periodos anteriores. Por ejemplo comparar las ventas de junio de 2009 con las ventas de junio de 2008, 2007, 2006, etc.... También es muy común hacer operaciones acumuladas con periodos móviles: por ejemplo las ventas de los últimos 6 meses o desde enero hasta el mes actual, etc...

Para todo ello MDX permite identificar y tratar de forma especial las dimensiones temporales, para poder aplicar las fórmulas que permiten obtener los cálculos que hemos mencionado en el párrafo anterior. A continuación nombramos algunas de las principales fórmulas relacionadas con las dimensiones temporales y su sintaxis.

### PeriodsToDate

Funcionalidad: Devuelve un conjunto de miembros del mismo nivel que un miembro determinado, empezando por el primer miembro del mismo nivel y acabando con el miembro en cuestión, de acuerdo con la restricción del nivel especificado en la dimensión de tiempo.

Sintaxis: <Conjunto> PeriodsToDate(<Nivel>, <Miembro>)

Ejemplo: *PeriodsToDate([Tiempo].[Mensual].[Semestre],[Tiempo].[Mensual].[Mes].[Agosto 2009]*

El siguiente ejemplo devolvería el conjunto de meses desde agosto de 2009 hasta el final del semestre es decir julio y agosto de 2009.

Existen funciones especiales para simplificar el uso de PeriodsToDate en sus versiones más comunes y son las siguientes:

**WTD(<Miembro>):** Devuelve los miembros de la misma semana del miembro especificado.

**MTD(<Miembro>):** Devuelve los miembros del mismo mes del miembro especificado.

**QTY(<Miembro>):** Devuelve los miembros del mismo trimestre del miembro especificado.

**YTD(<Miembro>):** Devuelve los miembros del mismo año del miembro especificado.

#### **ParallelPeriod**

Funcionalidad: Devuelve un miembro de un periodo anterior en la misma posición relativa que el miembro especificado.

Sintaxis: <Miembro> ParallelPeriod(<Nivel>, <Expresión numerica>, <Miembro>)

Ejemplo: *ParallelPeriod ([Tiempo].[Mes].[Semestre], 3, [Tiempo].[Mensual].[Mes].[Junio 2009])* El siguiente ejemplo devolvería diciembre 2007 que es el equivalente al último mes del semestre retrocediendo 3 semestres en el tiempo.

Hay más funciones de tiempo pero con estas sólo pretendo ilustrar el concepto y la utilidad de estas funciones. Cada uno deberá buscar la función que necesite para realizar sus cálculos y funcionalidades en sus cubos.

#### **4.- Miembros calculados y formulas complejas.**

Una de las funcionalidades más potentes que ofrece el lenguaje MDX es la posibilidad de realizar cálculos complejos tanto dinámicos (en función de los datos que se están analizando en ese momento) como estáticos. Los cubos multidimensionales trabajan con medidas (del inglés measures) y con miembros calculados (calculated members). Las medidas son las métricas de la tabla de hechos a las que se aplica una función de agregación (count, distinct count, sum, max, avg, etc...), en otras palabras son las métricas del negocio en estado puro. Los miembros calculados son tratados como métricas a la hora de explotar los cubos, sin embargo son mucho más potentes que estas. Un miembro calculado es una métrica que tiene como valor el resultado de la aplicación de una fórmula que puede utilizar todos los elementos disponibles en un cubo, así como otras funciones de MDX disponibles. Veamos un ejemplo sencillo.

Imaginamos que tenemos dos métricas en nuestro cubo que son: [Medidas].[importe ventas] y [Medidas].[Número de artículos vendidos].

Podríamos obtener el precio medio de artículo utilizando un miembro calculado con la siguiente fórmula:

[Medidas].[precio medio articulo]=[Medidas].[importe ventas] / [Medidas].[Número de artículos vendidos]

Voilà!!! Este es nuestro primer miembro calculado.

Los miembros calculados son realmente muy potentes pues podemos utilizar expresiones condicionales para mostrar diferente información en función del nivel o las dimensiones que estamos visualizando. Un segundo ejemplo algo más complejo:

```
iif([Tiempo.mensual].CurrentMember.Level IS  
[Tiempo.mensual].[Mes],[Measures].[promedio mensual],null)
```

En el ejemplo anterior utilizamos una función condicional de MDX llamada *iif* que nos permite asignar valor a un miembro calculado sólo cuando estamos en un nivel determinado de una dimensión. Concretamente este miembro calculado sólo tiene valor cuando estamos en el nivel *Mes* de la dimensión temporal, en cualquier otro caso el valor es nulo.

Acabamos el capítulo con otro ejemplo algo más complejo.

```
<CalculatedMember name="Criticidad 6" dimension="Measures">  
  <formula>  
    sum({ParallelPeriod([tiempo].[periodo],1,  
[tiempo].CurrentMember).Lag(5):ParallelPeriod([tiempo].[periodo],1,  
[tiempo].CurrentMember)},[Importe Ventas])  
  </formula>  
  
  <CalculatedMemberProperty name="FORMAT_STRING" expression="IIf(  
    sum({ParallelPeriod([tiempo].[periodo],1,  
[tiempo].CurrentMember).Lag(5):ParallelPeriod([tiempo].[periodo],1,  
[tiempo].CurrentMember)},[Importe  
Ventas])>0,'|##,##%|arrow=up','|##,##%|arrow=down')"/>  
  </CalculatedMember>
```

Aquí podemos ver un ejemplo de como se define una formula más compleja y como se da un formato especial en función del valor a esa formula.

La fórmula está utilizando una función de agregación para sumar la métrica [Importe Ventas] haciendo el acumulado de los últimos 5 valores del nivel ( 5 años, 5 meses o 5 semanas) utilizando la función de tiempo ParallelPeriod con el intervalo de los últimos 5 miembros (con ayuda de la funcion Lag(5))

El formato se define en la propiedad FORMAT\_STRING y consiste en una flecha verde o roja en función de si la cantidad es negativa o positiva utilizando de nuevo la función if y las propiedades de formato como arrow=up y arrow=down que son propias de la implementación de mondrian.

Espero que estos ejemplos os hayan servido para entender la utilidad y la potencia de los miembros calculados.

## 5.- Bibliografía. Para saber más

Algunas webs consultadas:

[www.mondrian.pentaho.org](http://www.mondrian.pentaho.org)

[http://en.wikipedia.org/wiki/Multidimensional\\_Expressions](http://en.wikipedia.org/wiki/Multidimensional_Expressions)

<http://msdn.microsoft.com/es-es/library/ms145595.aspx>

Un libro de referencia:

Libro: [MDX solutions with Microsoft SQL Server Analysis Services 2005](#) and Hyperion Essbase, George Spofford and others, Editorial Wiley