



Es el lenguaje diseñado para crear objetos y consultar modelos tabulares. Es una colección de funciones, operadores y constantes que se pueden utilizar en una fórmula o expresión para calcular y devolver uno o más valores que sirven para enriquecer el modelo de datos.

Modelo Tabular → Es un tipo de modelo de datos que organiza la información en una estructura de tabla, similar a una hoja de cálculo o una base de datos relacional. En este modelo, los datos se almacenan en filas y columnas, donde cada fila representa una entidad o una observación única, y cada columna representa un atributo o una característica de esa entidad.

Sus principales características son:

- Sirve para crear medidas, medidas rápidas, columnas y tablas.
- Como resultado se puede obtener un objeto de tabla o un valor escalar.
- Las funciones se pueden anidar.
- Cada función de DAX tiene requerimientos de tipos de datos de entrada y de salida.
- Admite el uso de variables.
- Permite agregar columnas calculadas y métricas al modelo, utilizando una sintaxis intuitiva.

Columna Calculada

- Los valores se calculan en base a la información de cada fila de una tabla (tiene contexto de fila)
- Agrega valores estáticos a cada fila de una tabla y los almacena en el modelo (lo que aumenta el tamaño del archivo)
- Se recalculan en la actualización de la fuente de datos
- Se utiliza principalmente como filas, columnas, segmentaciones o filtros.

Métricas

- Los valores se calculan en base a la información de cualquier filtro en el informe (tiene contexto de filtro)
- No crea nuevos datos en las propias tablas. No aumenta el tamaño del archivo
- Se vuelven a calcular en respuesta a cualquier cambio en los filtros dentro del informe
- Casi siempre se utiliza dentro del campo de valores de un objeto visual.

Nombre de la métrica

Nombre de la tabla

Contador Sujetos = **DISTINCTCOUNT**(Ventas[Identificación])

Función

Nombre de la columna



Operadores DAX

DAX igual que todos los lenguajes usa sus operadores para crear expresiones que comparan valores, realizan cálculos aritméticos o trabajan con cadenas.

Tipos de Operadores

Operadores aritméticos

Operadores aritméticos	Significado	Ejemplo
+ (signo más)	Suma	3+3
- (signo menos)	Resta o inversión del signo	3-1-1
* (asterisco)	Multiplicación	3*3
/ (barra diagonal)	División	3/3
^ (símbolo de intercalación)	Exponenciación	16^4

Operadores de comparación

Operadores de comparación	Significado	Ejemplo
=	Igual a	[Region] = "USA"
>	Mayor que	[Sales Date] > "Jan 2009"
<	Menor que	[Sales Date] < "Jan 1 2009"
>=	Mayor o igual que	[Amount] >= 20000
<=	Menor o igual que	[Amount] <= 100
<>	No igual a	[Region] <> "USA"

Operadores de concatenación

Operador de texto	Significado	Ejemplo
& ("Y comercial")	Conecta, o concatena, dos valores para generar un valor de texto continuo.	[Region] & ", " & [City]



Operadores lógicos

Operador de texto	Significado	Ejemplos
&& (doble Y comercial)	Crea una condición AND entre dos expresiones que tienen ambas un resultado booleano. Si ambas expresiones devuelven TRUE, la combinación de las expresiones también devuelve TRUE; de lo contrario, la combinación devuelve FALSE.	<code>([Region] = "France") && ([BikeBuyer] = "yes"))</code>
(doble barra vertical)	Crea una condición OR entre dos expresiones lógicas. Si alguna de las expresiones devuelve TRUE, el resultado es TRUE; solo cuando ambas expresiones son FALSE, el resultado es FALSE.	<code>(([Region] = "France") ([BikeBuyer] = "yes"))</code>
IN	Crea una condición OR lógica entre cada fila que se compara con una tabla. Nota: En la sintaxis del constructor de tabla se emplean llaves.	<code>'Product'[Color] IN { "Red", "Blue", "Black" }</code>



Prioridades de operadores

Operador	Descripción
^	Exponenciación
-	Inversión del signo (como en -1)
* y /	Multipliación y división
+ y -	Suma y resta
&	Conecta dos cadenas de texto (concatenación)
=,=,<,>,<=,>=,<>,IN	De comparación
NOT	NOT (operador unario)

Páginas de referencia DAX

- [Referencias DAX Microsoft](#)
- [Cartas DAX](#)

Formato de fórmulas, comentarios y buenas prácticas en DAX

Entre las buenas prácticas para crear expresiones con DAX podemos indicar las siguiente:

- Crear saltos de línea y sangrado en las expresiones facilita la legibilidad (Para crear un saldo de línea mantenemos presionada la tecla *ALT* y presionamos la tecla *Enter*).

```
ACTIVOS INMOVILIZADOS =  
ABS (  
    CALCULATE (  
        [Saldo del Período],  
        FILTER (  
            ResumenCuentas,  
            ResumenCuentas[Cuenta] = "ACTIVOS"  
            && ResumenCuentas[Detalle cuenta] IN {"ACTIVO INTANGIBLE", "DEP. ACUMULADA"}  
        )  
    )  
)
```



- Es recomendable escribir comentarios para documentar la funcionalidad de nuestra formula.
- Para hacer un comentario por línea lo haremos con doble barra con ello lo que venga después de “//” se pintará de color verde. Ejemplo.

```
PATRIMONIO =  
// Fórmula de Patrimonio  
ABS(CALCULATE([Saldo del Período],FILTER(ResumenCuentas,ResumenCuentas[Cuenta] = "PATRIMONIO")))
```

- Si quisiera comentar varias líneas, usaremos al inicio del comentario una barra y un asterisco (/*) y al final del comentario un asterisco y una barra (*/)

```
PATRIMONIO =  
/*  
    Descripción    = Fórmula de Patrimonio  
    Elaborada por = Carlos Azabache  
*/  
ABS(CALCULATE([Saldo del Período],FILTER(ResumenCuentas,ResumenCuentas[Cuenta] = "PATRIMONIO")))
```

- Trabajar con variables, las variables se evaluarán de acuerdo al contexto actual y una vez evaluadas, funcionan como valores constantes en el momento que se utilizan.

Para trabajar con variables se utiliza dos palabras reservadas que se colorearán de azul (VAR y RETURN)

```
NOF VS FM 2 =  
var _NOF = [Cuentas por Cobrar]+[Existencias] - [Cuentas por Pagar]  
var _FM  = [ACTIVOS CORRIENTES] - [PASIVOS CORRIENTES]  
RETURN  
_NOF - _FM
```

El uso de variables hará que nuestras formulas funcionen de forma eficiente ya que:

- Si determinado cálculo se utiliza más de una vez, se optimiza la consulta.
- Son fáciles de configurar.
- Permite escribir código más legible.
- Las expresiones se pueden evaluar mejor, ya que deja ver toda la lógica de la expresión.

Laboratorio 1

- Laboratorio 1 – Parte 1
- Laboratorio 1 – Parte 2