

Vista de Consultas DAX

**DEFINE Medidas(MEASURE), Variables (VAR),
Tablas (TABLE), Columnas calculadas (en contexto
de consulta). EVALUATE**

Jose Ignacio González Gómez

Departamento de Economía Contabilidad y Finanzas - Universidad de La Laguna

www.jggomez.eu

V.2.6

Objetivo y estructura de la vista de consulta

- El bloque DEFINE y EVALUATE
- Acceso a mostrar todas las medidas de la tabla/modelo. Consultas Rápidas

Ejemplos

- DEFINE con MEASURE (Medidas)
- DEFINE con VAR (Variables)
- DEFINE con TABLE (Tablas)
- DEFINE Columnas calculadas (en contexto de consulta)

Notas de uso y buenas prácticas

Errores comunes y cómo evitarlos

Objetivo de la vista de consulta DAX

Esta opción nos permite:

- Permite crear cálculos temporales sin alterar el modelo.
- Facilita pruebas y depuración en la ventana de consultas.
- Es la base para generar scripts DAX más complejos.

The screenshot shows the Power BI ribbon with the 'Inicio' tab selected. Below the ribbon is a toolbar with various icons for clipboard operations, search, and command palettes. The main area displays a DAX script. The script starts with a 'DEFINE' block (line 1) and an 'EVALUATE' block (line 2). A red arrow points from the 'DAX' icon in the left sidebar to the 'DAX' keyword in the script. The results pane at the bottom shows a single row of data with one column labeled '[Importe]' containing the value 30458. The bottom navigation bar includes tabs for 'Consulta 1' and 'Consulta 2', and a '+' button.

A circled number 1 is placed over the 'DEFINE' keyword. A circled number 2 is placed over the 'EVALUATE' keyword. A circled letter B is placed over the results table. A circled letter C is placed over the bottom navigation bar.

```
1 DEFINE
2   MEASURE 'Ventas'[Importe] = SUMX(Ventas,Ventas[Cantidades]*Ventas[Precio])
3
4 EVALUATE
5   SUMMARIZECOLUMNS(
6     "Importe", [Importe]
7   )
```

El bloque comienza con **DEFINE** donde se declara los elementos que se van a usar en la consulta antes de ejecutar el cálculo principal. **EVALUATE** es una instrucción que se utiliza para devolver el resultado de una tabla.

DEFINE, ¿para qué sirve?

DEFINE permite definir objetos temporales como (Estos objetos se crean solo para la duración de la consulta, no se guardan en el modelo)

- **Medidas** (MEASURE)
- **Variables** (VAR)
- **Tablas** (TABLE)
- **Columnas calculadas** (en contexto de consulta)

The screenshot shows the Power BI DAX Editor interface. On the left, there's a vertical toolbar with icons for Report, Model, DAX, and TMDL. The main area has a status bar at the top with a note about saving DAX queries to the model. Below that is a toolbar with 'Ejecutar' (Run) and 'Actualizar modelo con cambios (0)' (Update model with changes (0)). The main workspace contains a DAX query:

```
1 DEFINE  
2     MEASURE 'Ventas'[TotalVentas] = SUM('Ventas'[Importe])  
3     VAR Limite = 1000  
4 EVALUATE  
5     FILTER(  
6         SUMMARIZECOLUMNS('Clientes'[Nombre], "Ventas", [TotalVentas]),  
7         [TotalVentas] > Limite  
8     )  
9     ORDER BY [TotalVentas] DESC
```

Aspectos básicos relacionados con la vista de consulta DAX

Acceso a mostrar todas las medidas de la tabla/modelo. Consultas Rápidas

En la vista de consultas DAX y seleccionando con el botón derecho una carpeta, accedemos a “**Consultas Rápidas**” que nos muestra la posibilidad de construir una tabla con todas las medidas del modelo así como acceder todas las fórmulas y darle formato.

The screenshot shows the Power BI DAX view interface. On the left, there's a code editor window containing DAX code:

```
1 DEFINE
2     MEASURE 'MisMedidas'[Beneficio Bruto] = [Ingresos Netos]-[Coste de Ventas]
3     MEASURE 'MisMedidas'[Comisiones Vendedores] = SUMX('Ventas y Costes', 'Venta'
4     MEASURE 'MisMedidas'[Coste Reciclado] = SUMX('Ventas y Costes', 'Ventas y Co
5
6 EVALUATE
7     SUMMARIZECOLUMNS(
8         "Beneficio Bruto", [Beneficio Bruto],
9         "Comisiones Vendedores", [Comisiones Vendedores],
10        "Coste Reciclado", [Coste Reciclado]
11    )
```

Below the code editor is a results table:

	[Beneficio Bruto]	[Comisiones Vendedores]	[Coste Reciclado]
1	735794.94	90068.59	4162.89

On the right, a context menu is open over a folder icon. The menu is titled “Datos” and has two tabs: “Tablas” (selected) and “Modelo”. The “Tablas” tab contains a search bar and a list of tables: “Cuentas”, “Comisiones y Reciclado”, and “Ventas y Costes”. A red arrow points to the “Consultas rápidas” option in the “Modelo” tab, which is highlighted with a red box. The “Consultas rápidas” option has a submenu with the following items:

- Mostrar las 100 primeras filas
- Mostrar estadísticas de columna
- Definir nueva medida
- Definir todas las medidas de esta tabla (highlighted with a red box)
- Definir todas las medidas de este modelo (highlighted with a red box)
- Consultas rápidas
- Actualizar
- Administrar relaciones
- Seleccionar columnas
- Seleccionar medidas
- Cambiar nombre
- Eliminar del modelo
- Ocultar en la vista de informes
- Mostrar todos
- Contraer todo
- Expandir todo

Con la opción 1 podemos dar formato a las medidas y con (2) quedan actualizadas. Ademas disponemos tambien de la opción de añadir comentarios.

The screenshot shows the Power BI DAX Editor interface. The top navigation bar includes 'Archivo', 'Inicio' (which is underlined), 'Ayuda', and 'Herramientas externas'. Below the navigation bar is a toolbar with icons for 'Pegar' (Paste), 'Cortar' (Cut), 'Copiar' (Copy), 'Formato' (Format), 'Comentario' (Comment), 'Quitar marca de comentario' (Remove comment), 'Buscar' (Search), 'Reemplazar' (Replace), 'Paleta de comandos' (Command palette), and 'Copilot'. A red arrow labeled '1' points to the 'Formato' icon. A red circle labeled '2' highlights the first line of the DAX script, which defines a measure named 'Beneficio Bruto'.

Las consultas DAX se guardarán en el modelo. No serán visibles cuando se publiquen en el servicio Power BI. [Más información](#).

Ejecutar Actualizar modelo con cambios (3)

```
1 DEFINE
2   Actualizar modelo: sobre escribir medida
3     MEASURE 'MisMedidas'[Beneficio Bruto] = [Ingresos Netos] - [Coste de Ventas]
4   Actualizar modelo: sobre escribir medida
5     MEASURE 'MisMedidas'[Comisiones Vendedores] = SUMX(
6       'Ventas y Costes',
7       'Ventas y Costes'[Ingresos Directos] * RELATED('Comisiones y Reciclado'[Comisiones Vendedores])
8     )
9     Actualizar modelo: sobre escribir medida
10    MEASURE 'MisMedidas'[Coste Reciclado] = SUMX(
11      'Ventas y Costes',
12      'Ventas y Costes'[Costes Directos] * RELATED('Comisiones y Reciclado'[Reciclado ])
13    )
14
15 EVALUATE
16   SUMMARIZECOLUMNS(
17     "Beneficio Bruto", [Beneficio Bruto],
18     "Comisiones Vendedores", [Comisiones Vendedores],
19     "Coste Reciclado", [Coste Reciclado]
```

Resultados Resultado 1 de 1 Copia

	[Beneficio Bruto]	[Comisiones Vendedores]	[Coste Reciclado]
1	735794.94	90068.59	4162.89

Ejemplo 1: Medida temporal para ventas totales

```
1 DEFINE  
2     MEASURE 'Ventas'[TotalVentas] = SUM('Ventas'[Importe])  
3  
4 EVALUATE  
5     ADDCOLUMNS(  
6         SUMMARIZECOLUMNS('Clientes'[Nombre]),  
7         "Ventas Totales", [TotalVentas]  
8     )
```

¿Qué hace?

- Define una **medida temporal** en la tabla Ventas [TotalVentas] que suma la columna 'Ventas'[Importe].
- Devuelve una tabla con cada cliente y su total de ventas.

Ejemplo 2: Medida temporal con filtro año

```
1 DEFINE  
2     MEASURE 'Ventas'[Ventas2024] =  
3         CALCULATE(SUM('Ventas'[Importe]), 'Calendario'[Año] = 2024)  
4  
5 EVALUATE  
6     ADDCOLUMNS(  
7         SUMMARIZECOLUMNS('Clientes'[Nombre]),  
8         "Ventas 2024", [Ventas2024]  
9     )
```

¿Qué hace?

- Define una medida [Ventas2024] que calcula ventas solo para el año 2024.
- Devuelve una tabla con clientes y sus ventas en ese año.

Ejemplo 1: Variable para filtrar por año

```
1 DEFINE
2     VAR AñoSeleccionado = 2024
3
4 EVALUATE
5     FILTER(
6         ADDCOLUMNS(
7             SUMMARIZECOLUMNS('Clientes'[Nombre]),
8             "Ventas", CALCULATE(SUM('Ventas'[Importe]), 'Calendario'[Año] =
AñoSeleccionado)
9         ),
10        [Ventas] > 5000
```

¿Qué hace?

- Define una variable AñoSeleccionado = 2024.
- Calcula ventas por cliente para ese año.
- Filtra clientes con ventas mayores a 5000.

Ejemplo 2: Variable para umbral

```
1 DEFINE
2     VAR Umbral = 10000
3
4 EVALUATE
5     FILTER(
6         ADDCOLUMNS(
7             SUMMARIZECOLUMNS('Clientes'[Nombre]),
8             "TotalVentas", SUM('Ventas'[Importe])
9         ),
10        [TotalVentas] >= Umbral
11    )
```

¿Qué hace?

- Define una variable Umbral = 10000.
- Calcula ventas totales por cliente.
- Filtra clientes con ventas mayores o iguales al umbral.

Ejemplo 1: Crear una tabla temporal con clientes VIP

```
1 DEFINE
2     TABLE ClientesVIP =
3         FILTER('Clientes', 'Clientes'[Categoria] = "VIP")
4
5 EVALUATE
6     ClientesVIP
```

¿Qué hace?

- Define una tabla temporal llamada ClientesVIP que contiene solo los clientes cuya categoría es "VIP".
- EVALUATE devuelve esa tabla.

Ejemplo 2: Crear una tabla temporal con clientes VIP

```
1 DEFINE
2     TABLE Ventas2024 =
3         FILTER('Ventas', 'Ventas'[Año] = 2024)
4
5 EVALUATE
6     SUMMARIZECOLUMNS(
7         'Clientes'[Nombre],
8         "Ventas 2024", CALCULATE(SUM('Ventas'[Importe]), Ventas2024)
9     )
```

¿Qué hace?

- Define una tabla Ventas2024 con todas las filas de ventas del año 2024.
- Luego, en EVALUATE, usa esa tabla como filtro para calcular ventas por cliente.

Ejemplo 1: Columna calculada para clasificar clientes

```
1 DEFINE  
2     COLUMN 'Clientes'[CategoriaVentas] =  
3         IF(SUM('Ventas'[Importe]) > 10000, "Premium", "Estándar")  
4  
5 EVALUATE  
6     ADDCOLUMNS(  
7         SUMMARIZECOLUMNS('Clientes'[Nombre]),  
8         "Categoría", 'Clientes'[CategoriaVentas]  
9     )
```

¿Qué hace?

- Define una columna calculada temporal llamada [CategoriaVentas] en la tabla Clientes.
- Clasifica al cliente como Premium si sus ventas superan 10.000, si no, Estándar.
- Devuelve una lista de clientes con su categoría.

Ejemplo 2: Columna calculada para margen

```
1 DEFINE  
2     COLUMN 'Ventas'[Margen] =  
3         'Ventas'[Importe] - 'Ventas'[Coste]  
4  
5 EVALUATE  
6     ADDCOLUMNS(  
7         SUMMARIZECOLUMNS('Ventas'[Producto]),  
8         "Margen Total", SUM('Ventas'[Margen])  
9     )
```

¿Qué hace?

- Define una columna [Margen] en la tabla Ventas que calcula la diferencia entre Importe y Coste.
- Luego, devuelve el margen total por producto.

- EVALUATE siempre devuelve una tabla; puedes construirla con ADDCOLUMNS, SUMMARIZECOLUMNS, FILTER, SELECTCOLUMNS, TOPN, etc.
- Usa ORDER BY, START AT y TOPN tras EVALUATE para ordenar o limitar el conjunto de resultados.
- Si necesitas reutilizar lógicas complejas o parámetros, considera DEFINE con VAR, MEASURE o TABLE; para pruebas simples, EVALUATE suele ser suficiente.
- Evita consultas que escaneen tablas muy grandes sin filtros; añade agregaciones o restricciones para mejorar rendimiento.
- Recuerda que las expresiones dentro de la consulta no modifican el modelo; son ad-hoc para la ejecución.

ERRORES COMUNES Y CÓMO EVITARLOS

- Olvidar ORDER BY tras calcular columnas cuando se necesita un orden específico.
- Usar nombres de columnas sin el nombre de tabla cuando hay ambigüedad (prefiera "Tabla"[Columna]).
- Intentar devolver una medida directamente: las consultas deben devolver tablas; proyecte la medida con ADDCOLUMNS.
- Filtrar con columnas que no están relacionadas en el modelo: verifique relaciones y direccionalidad.