

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)

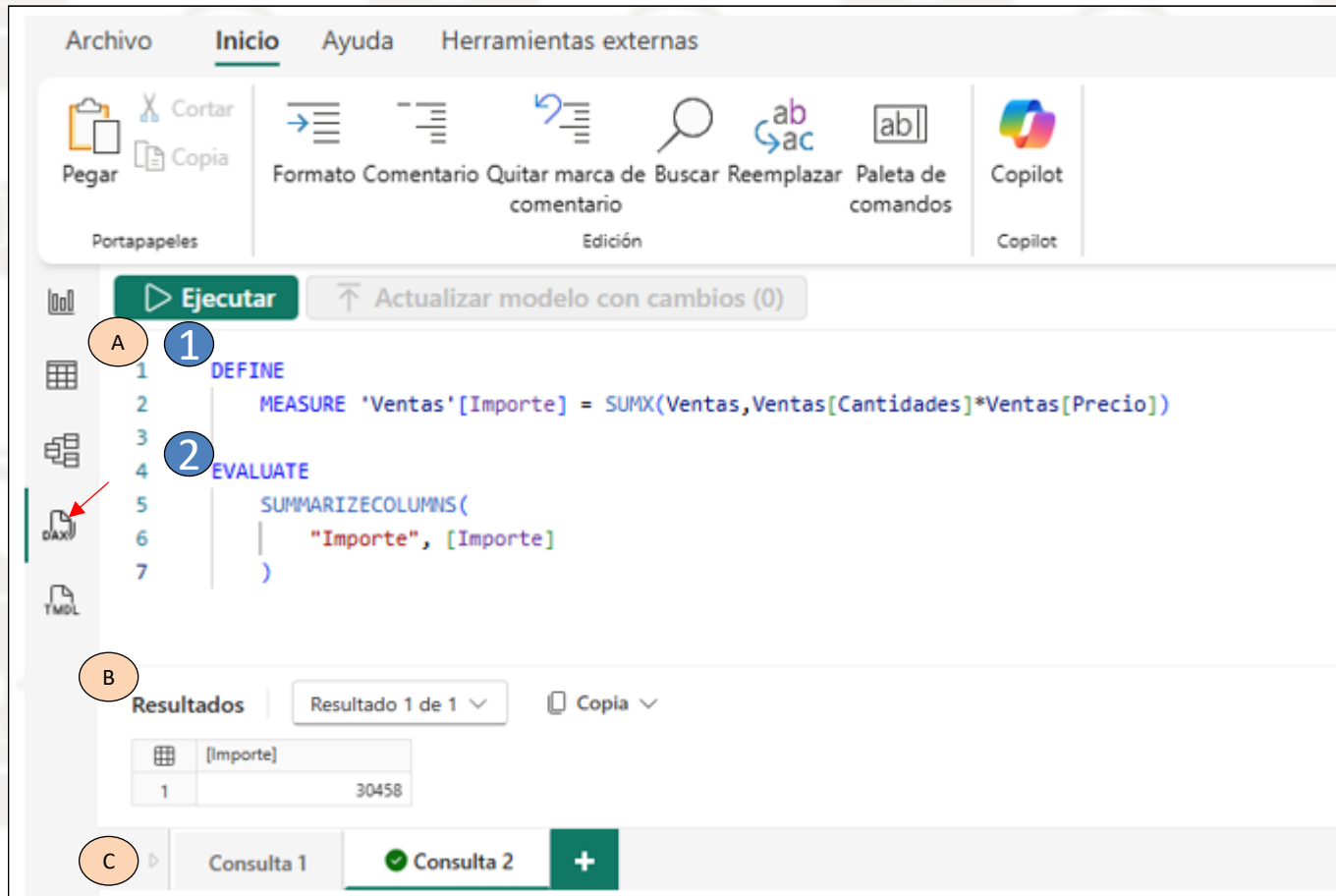
Notas de uso y buenas prácticas

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.




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)



 Las consultas DAX se guardarán en el modelo No serán visibles cuando se publiquen en el servicio Power BI.



 Ejecutar

 Actualizar modelo con cambios (0)



```
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 boton derecho una carpeta, accedemos a “**Consultas Rápidas**” que nos muestra la posibilidad de construir una tabla con todas las medidas del modelo asi como acceder todas las fórmulas y darle formato.

The screenshot displays the Microsoft Power BI Desktop interface. The top ribbon shows the 'Inicio' (Home) tab, with the 'Formato' (Format) button highlighted by a red box. The 'Consultas Rápidas' (Quick Queries) menu is open, showing options like 'Mostrar las 100 primeras filas' (Show first 100 rows), 'Mostrar estadísticas de columna' (Show column statistics), 'Definir nueva medida' (Define new measure), 'Definir todas las medidas de esta tabla' (Define all measures of this table), and 'Definir todas las medidas de este modelo' (Define all measures of this model). The 'Definir todas las medidas de este modelo' option is highlighted by a red box. A red arrow points to the 'Consultas rápidas' (Quick Queries) option in the 'Datos' (Data) pane on the right.

The DAX query editor shows the following code:

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

The 'Resultados' (Results) pane shows the following table:

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

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

ArchivoInicioAyudaHerramientas externas

Cortar

Pegar

Copia

Formato

Comentario

Quitar marca de comentario

1

Buscar

Reemplazar

Paleta de comandos

Copilot

Copilot

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

Actualizar modelo: sobrescribir medida

2

MEASURE 'MisMedidas'[Beneficio Bruto] = [Ingresos Netos] - [Coste de Ventas]

Actualizar modelo: sobrescribir medida

3

MEASURE 'MisMedidas'[Comisiones Vendedores] = SUMX(

4

'Ventas y Costes',

5

'Ventas y Costes'[Ingresos Directos] * RELATED('Comisiones y Reciclado'[Comisiones Vendedores])

6

)

Actualizar modelo: sobrescribir medida

7

MEASURE 'MisMedidas'[Coste Reciclado] = SUMX(

8

'Ventas y Costes',

9

'Ventas y Costes'[Costes Directos] * RELATED('Comisiones y Reciclado'[Reciclado])

10

)

11

EVALUATE

12

SUMMARIZECOLUMNS(

13

"Beneficio Bruto", [Beneficio Bruto],

14

"Comisiones Vendedores", [Comisiones Vendedores],

15

"Coste Reciclado", [Coste Reciclado]

16

)

17

)

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.

EJEMPLOS

DEFINE con VAR (Variables)

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.

EJEMPLOS

DEFINE con TABLE (Tablas)

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.