



---

[www.fcervantes.es](http://www.fcervantes.es)

[francisco@fcervantes.es](mailto:francisco@fcervantes.es)



---

# Curso Power BI Avanzado

Impartido por Francisco Guillermo Cervantes Medina

## Club de Marketing de La Rioja

18, 25 y 31 de octubre de 2024



**Club de  
Marketing  
de La Rioja**

TU SOCIO EN GESTIÓN AVANZADA



# Contenidos

---

1. Modelo de datos
2. Transformación de datos
3. DAX
4. Modelo de datos – Ambigüedad

# Modelo de datos



# Modelo de datos: Relaciones entre tablas

---

Para poder establecer una relación entre estas tablas, es necesario contar con un elemento común entre ellas.

Comercial  
Nombre cliente  
Comercial

Nombres clientes  
Código cliente  
Nombre  
Razón Social

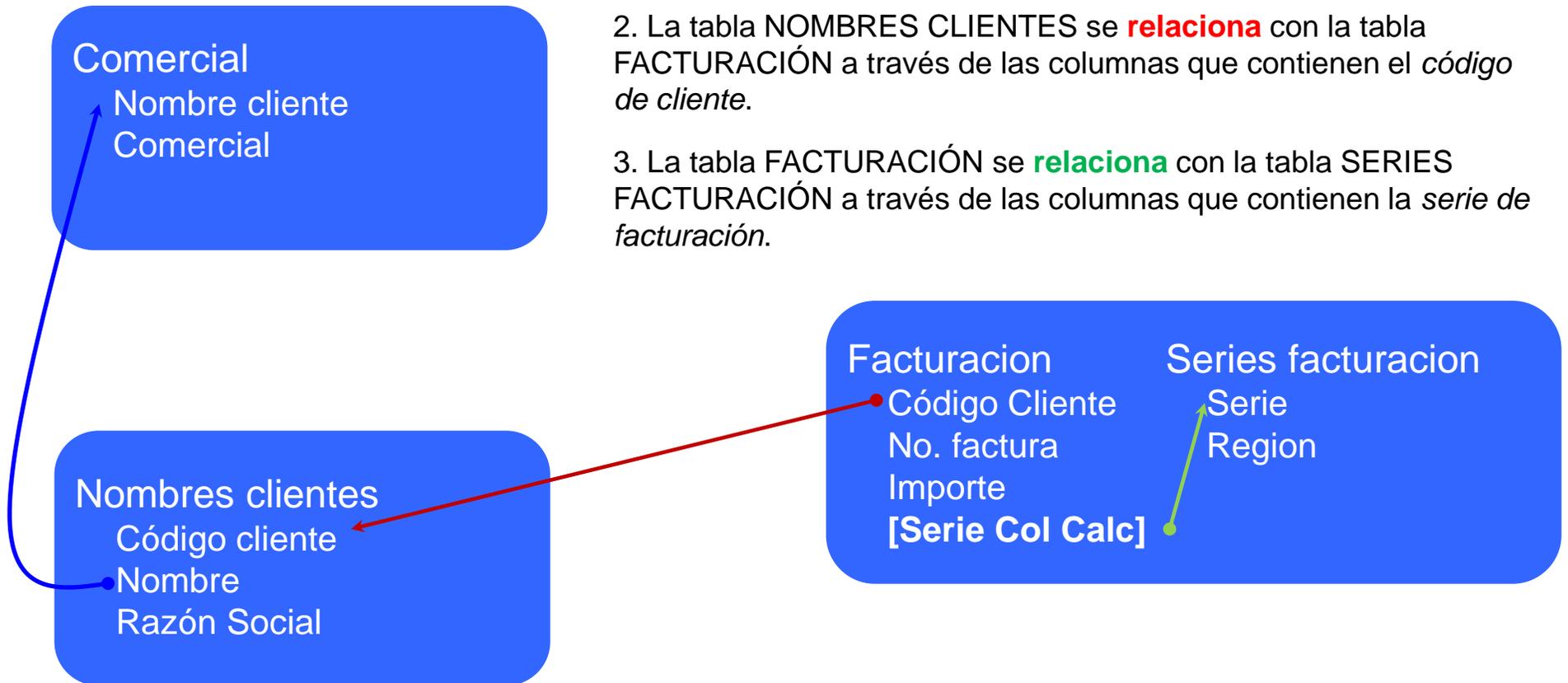
Facturacion	Series facturacion
Código Cliente	Serie
No. factura	Region
Importe	
<b>[Serie Col Calc]</b>	

# Modelo de datos: Relaciones entre tablas

1. La tabla COMERCIAL y la tabla NOMBRES CLIENTES se **relacionan** a través de las columnas que contienen el *nombre del cliente*.

2. La tabla NOMBRES CLIENTES se **relaciona** con la tabla FACTURACIÓN a través de las columnas que contienen el *código de cliente*.

3. La tabla FACTURACIÓN se **relaciona** con la tabla SERIES FACTURACIÓN a través de las columnas que contienen la *serie de facturación*.



# Modelo de datos: Relaciones entre tablas

Alexander Calder, 'Guava', 1955  
 © 2019 Calder Foundation, New York /  
 VEGAP, Santander



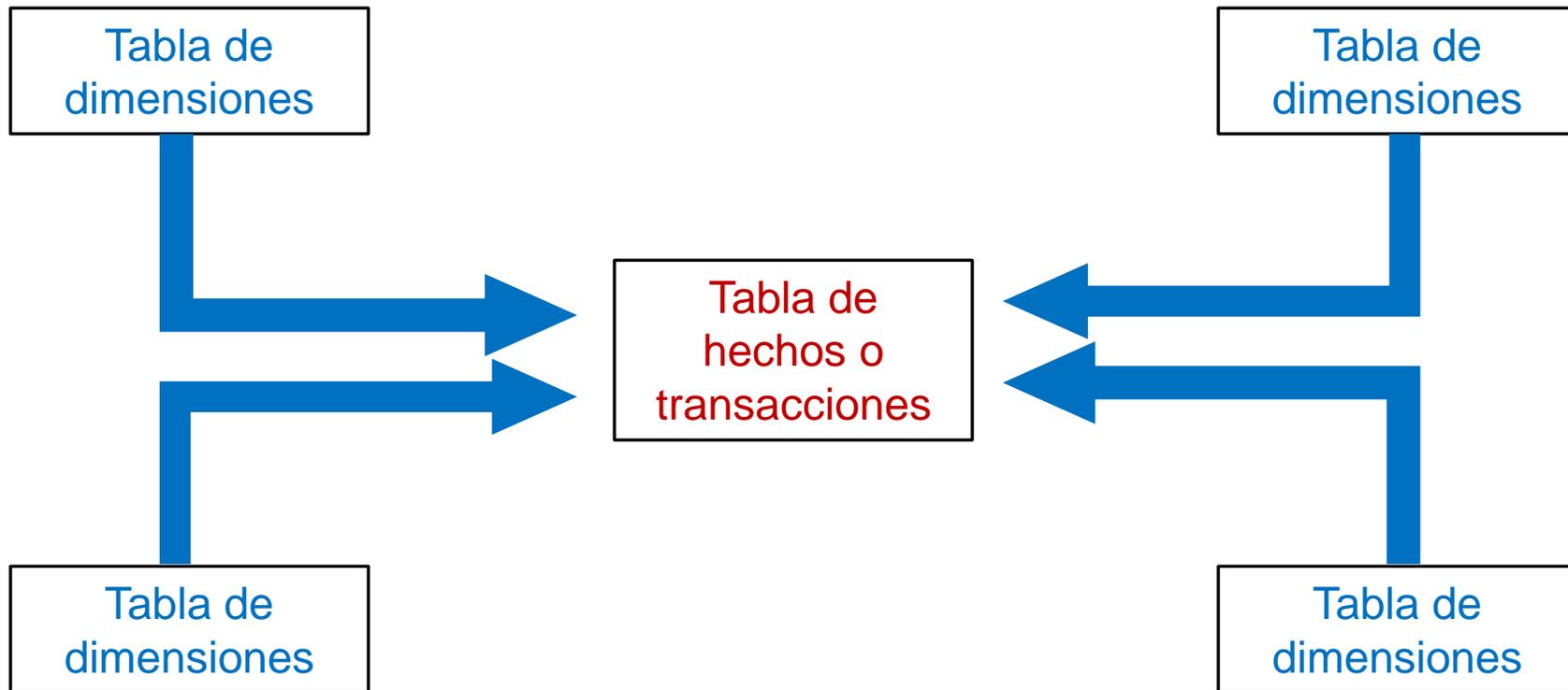
# Modelo de datos: Relaciones entre tablas



# Tipos de modelos de datos

---

## Modelo tipo “Estrella”



Todas las tablas están conectadas a una tabla de hechos o transacciones.

# Dirección el filtro cruzado

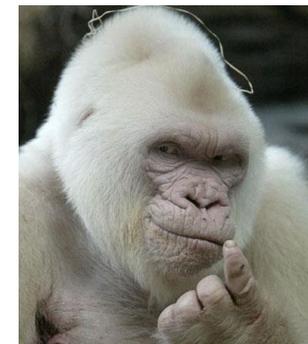
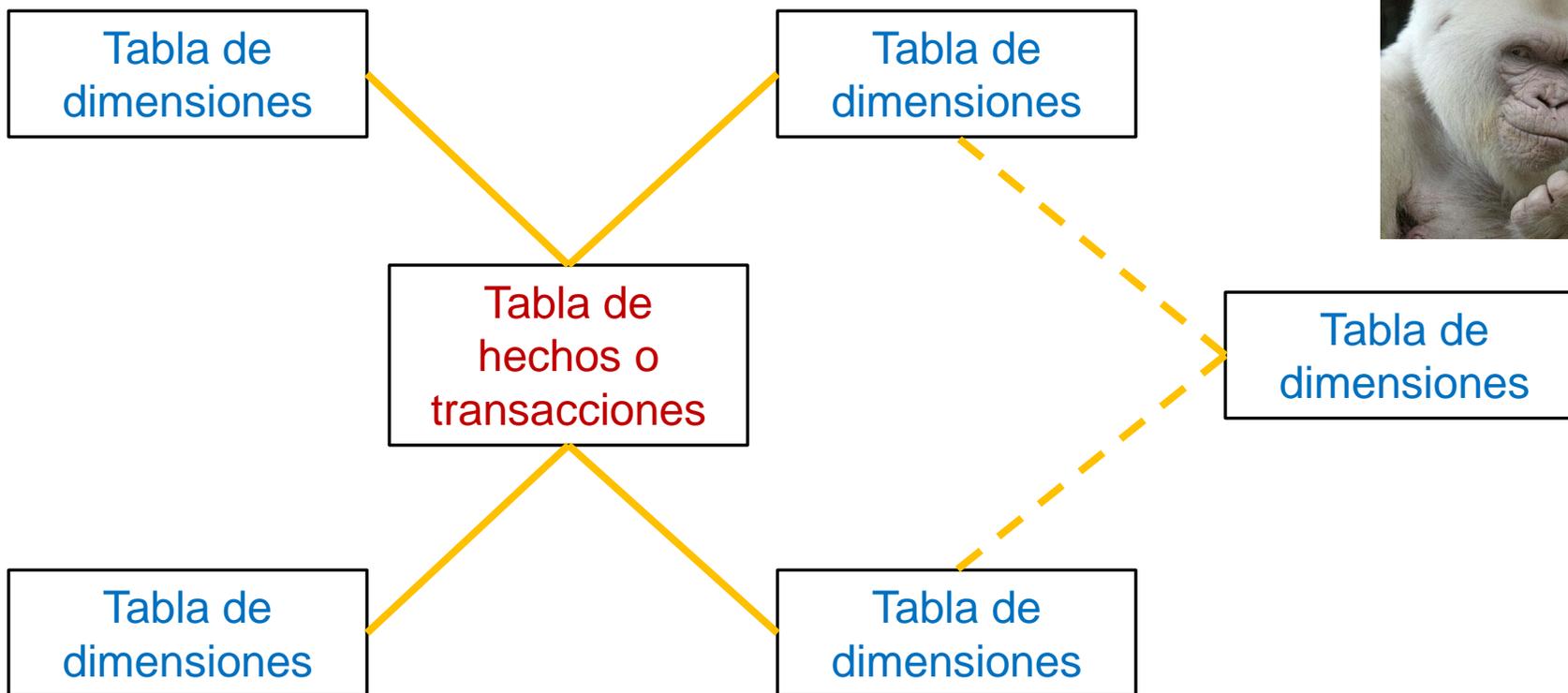
## Modelo tipo “Estrella”



En un modelo de “Estrella” todas las tablas de dimensiones están **directamente** conectadas a un nodo central que es la tabla de hechos.

# Tipos de modelos de datos

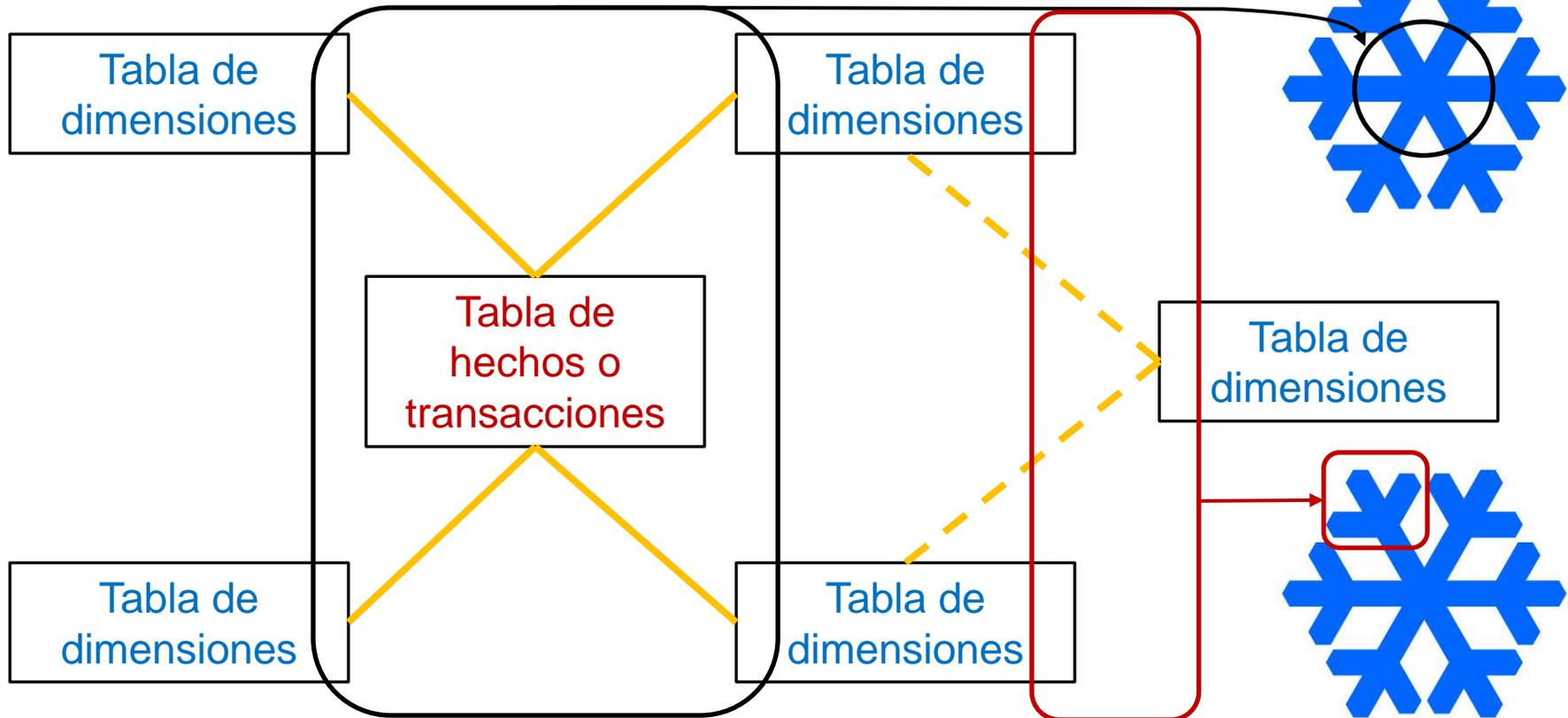
## Modelo tipo “copo de nieve”



En un modelo tipo “Copo de nieve” al menos una tabla de dimensiones **no** está conectada **directamente** a la tabla de hechos.

# Tipos de modelos de datos

## Modelo tipo “copo de nieve”



En un modelo tipo “Copo de nieve” al menos una tabla de dimensiones **no** está conectada **directamente** a la tabla de hechos.

# Modelo de datos: Relaciones entre tablas

**Editar relación**

Permite seleccionar tablas y columnas relacionadas.

Nombres clientes

Codigo cliente	Nombre	Razón Social
1	Los lunes	ABC, S.L.
2	Los martes	DEF, S.A.
3	Pérez escayolistas	Juan Pérez autónomo

Facturacion

Codigo cliente	No. Factura	Importe
1	A-512	1000
2	B-310	2000
3	A-340	3000

Cardinalidad: Uno a uno (1:1)

Dirección del filtro cruzado: Ambas

Activar esta relación

Asumir integridad referencial

Aceptar Cancelar

Para que pueda haber una relación entre dos tablas, éstas tienen que tener un campo (columna) común.

Una relación entre dos tablas en Power BI se define por 2 parámetros:

**1. Cardinalidad:** Definida, a su vez, por la *repetición de elementos* en las columnas que relacionan a las dos tablas.

**2. Dirección del filtro cruzado:** Qué tablas pueden filtrar a otras en el modelo. Es decir, cómo o hacia dónde “fluye” el filtro de una tabla a otra.



# Cardinalidad

Los elementos en la columna que se relaciona con la otra tabla son:

Tabla <sub>1</sub>		Tabla <sub>2</sub>		Tipo de tabla	Cardinalidad
Únic.	Rep.	Únic.	Rep.		
x		x		Tablas 1 y 2 → Dimensiones	Uno a Uno (1:1)
x			x	Tabla 1 → Dimensiones Tabla 2 → Hechos	Uno a Varios (1:*)
	x		x	Tablas 1 y 2 → Hechos	Varios a Varios (*:*)
	x	x		Tabla 1 → Hechos Tabla 2 → Dimensiones	Varios a Uno (*:1)

Las relaciones 1:1 pueden generar ineficiencias.

Las relaciones \*: \* pueden generar ambigüedad.

Lo mejor es evitarlas **en la medida de lo posible**.



# Cardinalidad

**Editar relación** ✕

Permite seleccionar tablas y columnas relacionadas.

Facturacion

Codigo cliente	No. Factura	Importe	ColCalc Serie Fact
1	A-512	1.000,00 €	A
2	B-310	2.000,00 €	B
3	A-340	3.000,00 €	A

Nombres clientes

Codigo cliente	Nombre	Razón Social
1	Los lunes	ABC, S.L.
2	Los martes	DEF, S.A.
3	Pérez escayolistas	Juan Pérez autónomo

Cardinalidad: **Varios a uno (\*:1)**

Dirección del filtro cruzado: **Única**

Activar esta relación

Aplicar filtro de seguridad en ambas direcciones

Asumir integridad referencial

Aceptar Cancelar



# Cardinalidad

Contiene o puede contener elementos repetidos en la columna que la relaciona con la otra tabla

*Un mismo cliente puede tener varias facturas. Por lo tanto, su código puede aparecer varias veces.*

→ **Tabla de hechos**

Contiene elementos únicos en la columna que la relaciona con la otra tabla

→ **Tabla de dimensiones**

*Un cliente sólo puede tener un código y viceversa, un código sólo puede corresponder a un cliente.*

### Editar relación

Permite seleccionar tablas y columnas relacionadas.

**Cardinalidad:**

Varios

a

Uno

Código cliente	No. Factura	Importe	ColCalc Serie Fact
1	A-512	1.000,00 €	A
2	B-310	2.000,00 €	B
3	A-340	3.000,00 €	A

Código cliente	Nombre	Razón Social
1	Los lunes	ABC, S.L.
2	Los martes	DEF, S.A.
3	Pérez escayolistas	Juan Pérez autónomo

Cardinalidad: **Varios a uno (\*:1)**

Dirección del filtro cruzado: **Única**

Activar esta relación

Aplicar filtro de seguridad en ambas direcciones

Asumir integridad referencial

**Aceptar** **Cancelar**



# Filtro cruzado

---

Filtro que va de una tabla a otra.

Es decir, si aplicamos el filtro a una tabla del modelo de datos, dicho filtro “viaja” a otras tablas del modelo.

**Un ejemplo esquemático...**



# Filtro cruzado

Categoría	
Categoría	Responsable línea producto
Cat A	Saúl Álvarez
Cat B	J.C. Chávez

Dimensiones

Hechos o Transacciones

Año
Año
2018
2019
2020
2021
2022

Dimensiones

Producto		
Cód. Prod.	Descripción	Categoría
1	Prod. A	Cat A
2	Prod. B	Cat B
3	Prod. C	Cat A

Dimensiones

Ventas		
Cód. Prod.	Uds. Vendidas	Año
2	10 uds	2018
1	3 uds	2019
3	8 uds	2021
2	6 uds	2018
3	15 uds	2022



# Filtro cruzado

Categoría	
Categoría	Responsable línea producto
Cat A	Saúl Álvarez
Cat B	J.C. Chávez

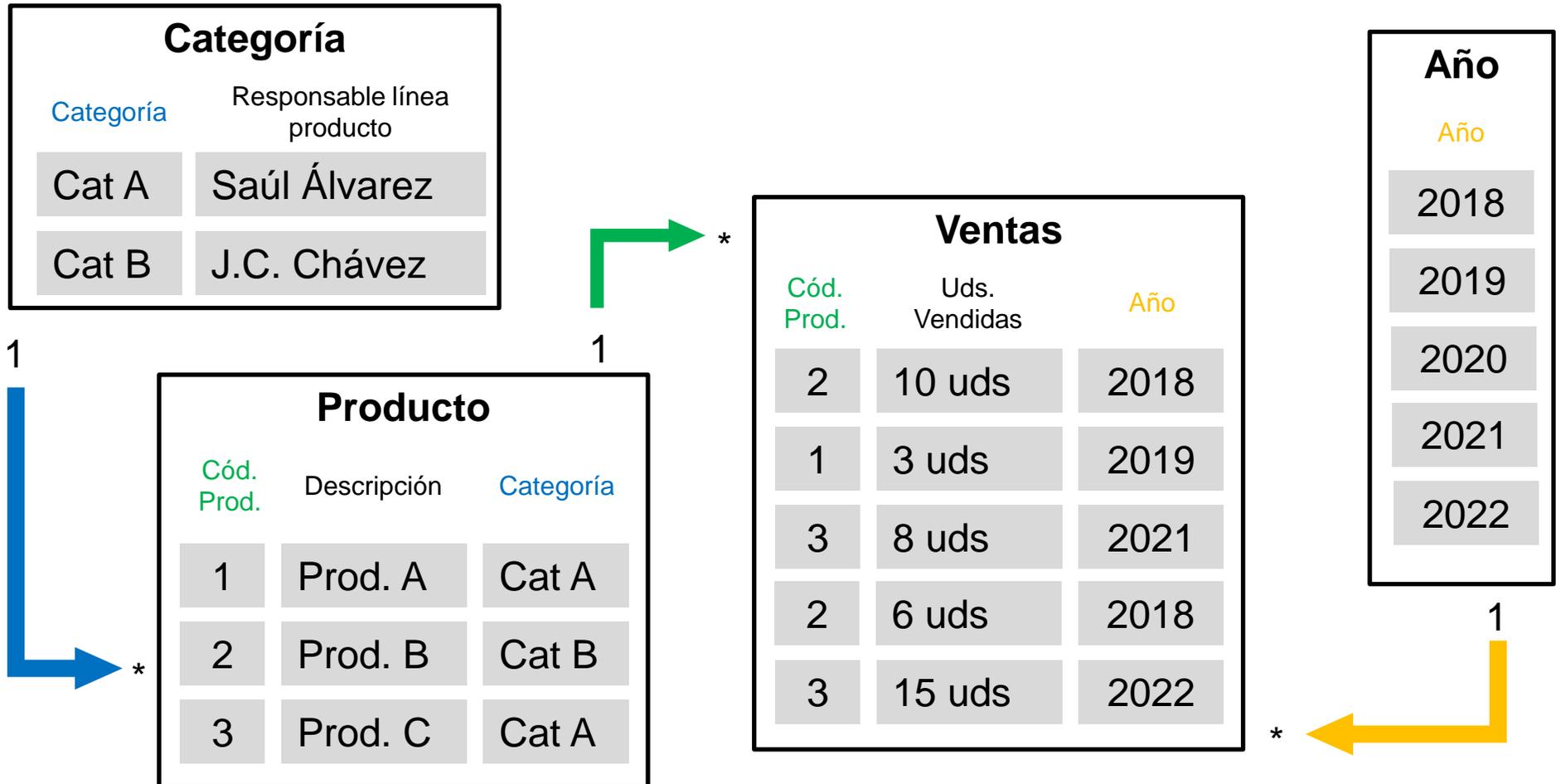
Producto		
Cód. Prod.	Descripción	Categoría
1	Prod. A	Cat A
2	Prod. B	Cat B
3	Prod. C	Cat A

Ventas		
Cód. Prod.	Uds. Vendidas	Año
2	10 uds	2018
1	3 uds	2019
3	8 uds	2021
2	6 uds	2018
3	15 uds	2022

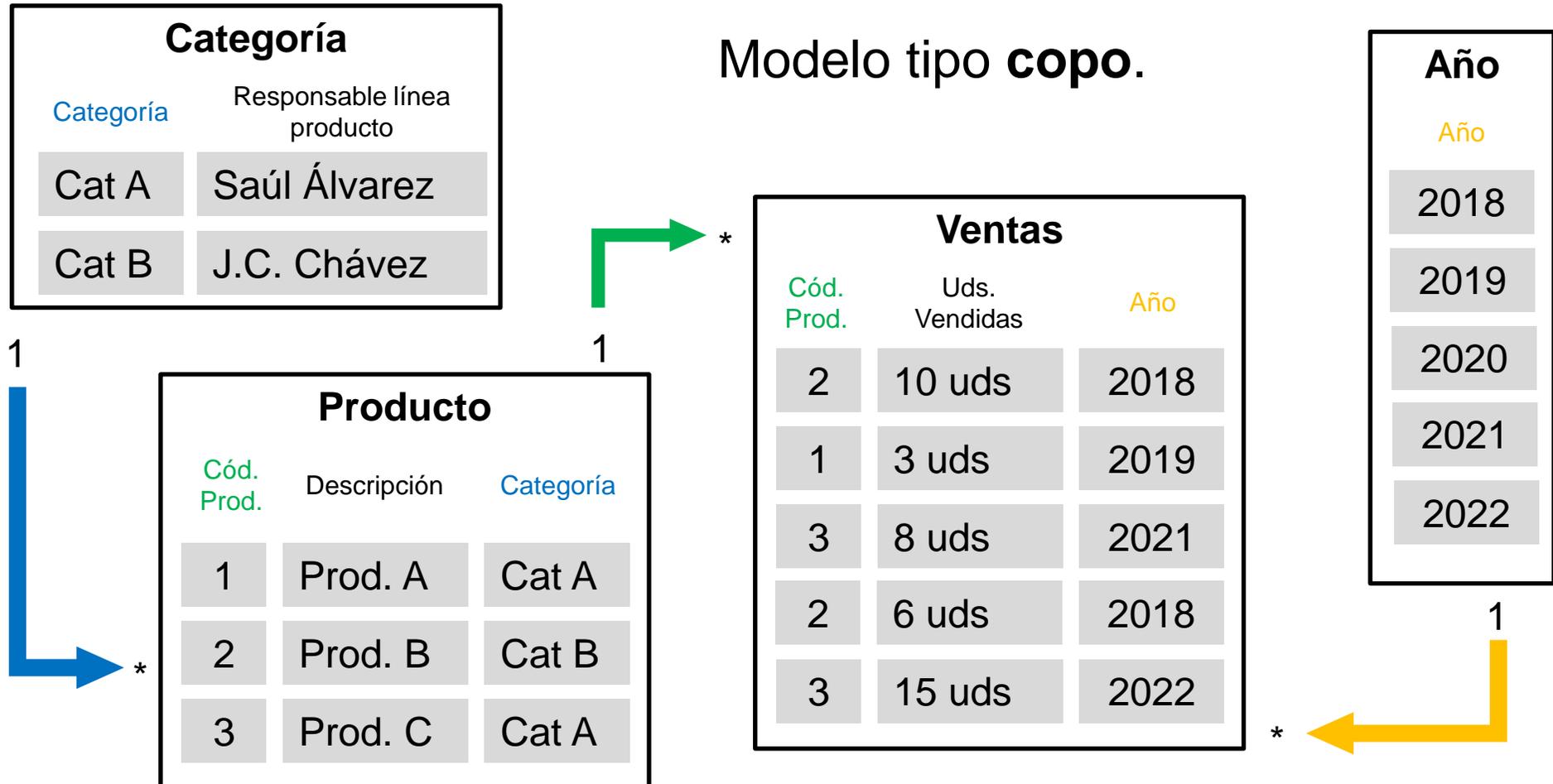
Año
Año
2018
2019
2020
2021
2022



# Filtro cruzado



# ¿Modelo tipo “copo” o “estrella” ?





# Filtro cruzado

Categoría	
Categoría	Responsable línea producto
Cat A	Saúl Álvarez
Cat B	J.C. Chávez

Quiero conocer las **ventas** de la **Cat. A** para el **2021**.

Año
Año
2018
2019
2020
2021
2022

Producto		
Cód. Prod.	Descripción	Categoría
1	Prod. A	Cat A
2	Prod. B	Cat B
3	Prod. C	Cat A

Ventas		
Cód. Prod.	Uds. Vendidas	Año
2	10 uds	2018
1	3 uds	2019
3	8 uds	2021
2	6 uds	2018
3	15 uds	2022



# Filtro cruzado

Categoría	
Categoría	Responsable línea producto
Cat A	Saúl Álvarez
Cat B	J.C. Chávez

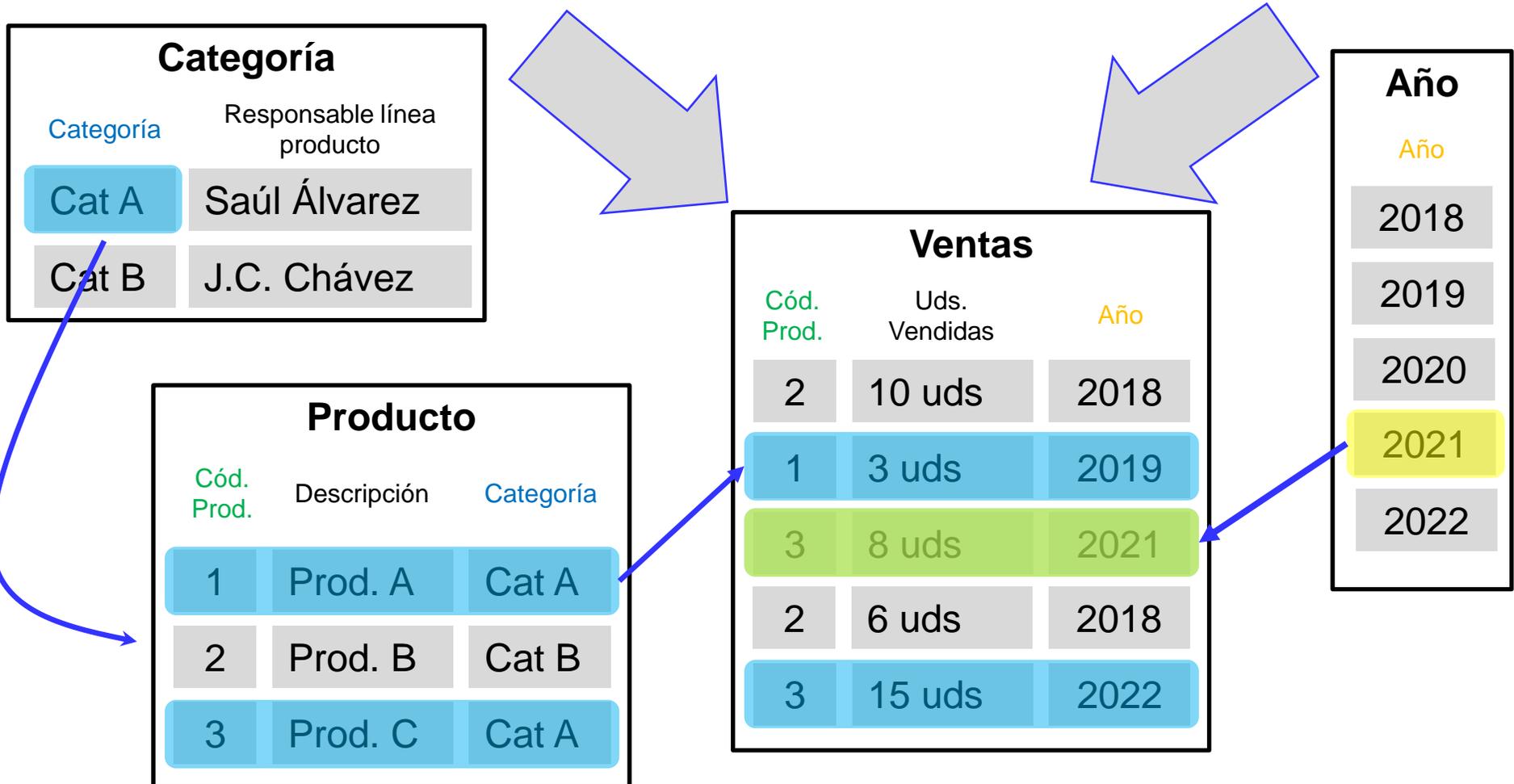
Quiero conocer las **ventas** de la **Cat. A** para el **2021**.

Año
Año
2018
2019
2020
2021
2022

Producto		
Cód. Prod.	Descripción	Categoría
1	Prod. A	Cat A
2	Prod. B	Cat B
3	Prod. C	Cat A

Ventas		
Cód. Prod.	Uds. Vendidas	Año
2	10 uds	2018
1	3 uds	2019
3	8 uds	2021
2	6 uds	2018
3	15 uds	2022

# Dirección del filtro cruzado



# Dirección del filtro cruzado

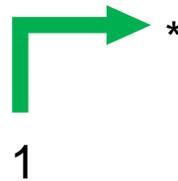
Categoría	
Categoría	Responsable línea producto
Cat A	Saúl Álvarez
Cat B	J.C. Chávez

Quiero saber a qué **responsables de línea** atribuir las ventas del **2018**.

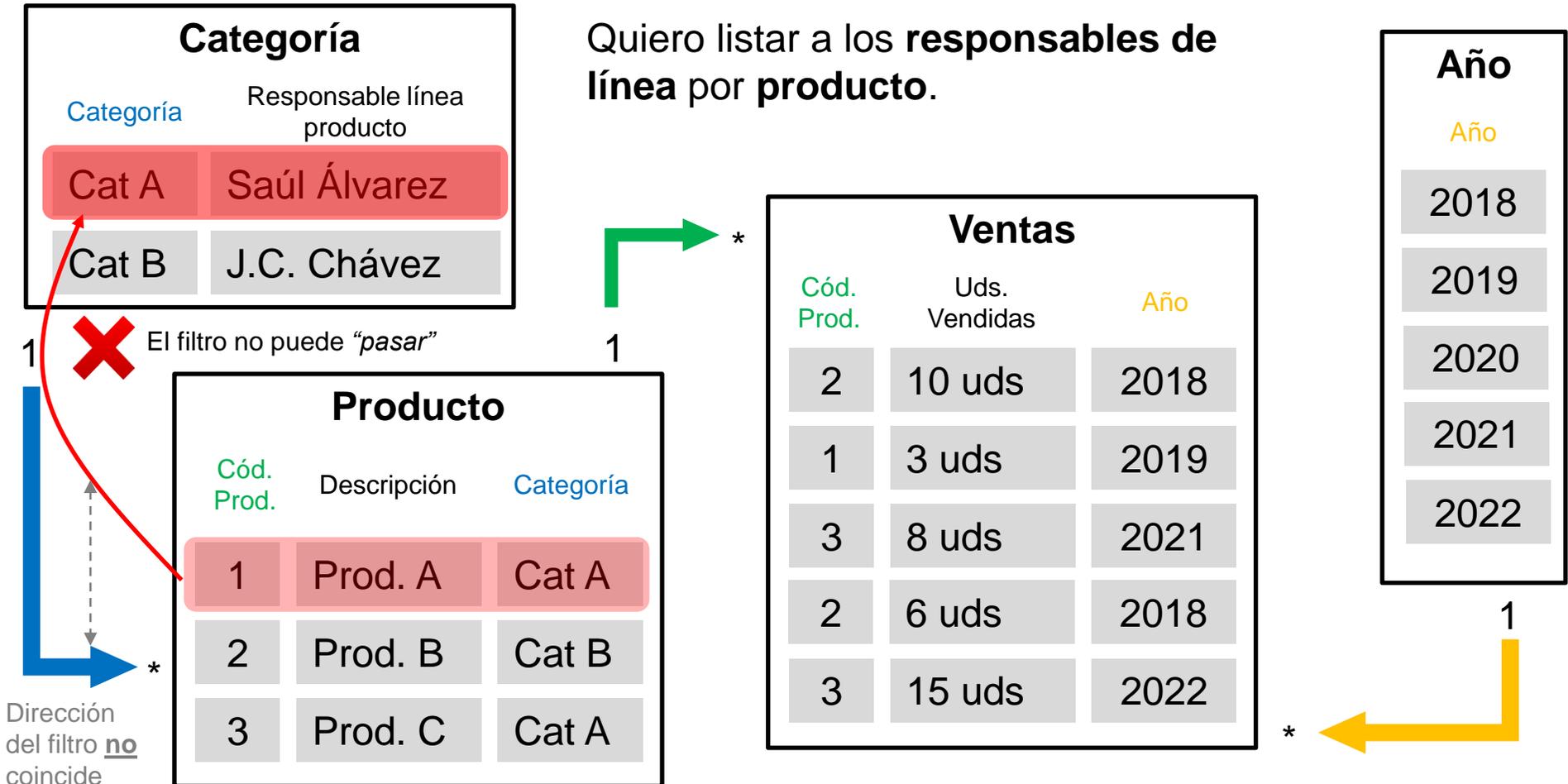
Año
Año
2018
2019
2020
2021
2022

Producto		
Cód. Prod.	Descripción	Categoría
1	Prod. A	Cat A
2	Prod. B	Cat B
3	Prod. C	Cat A

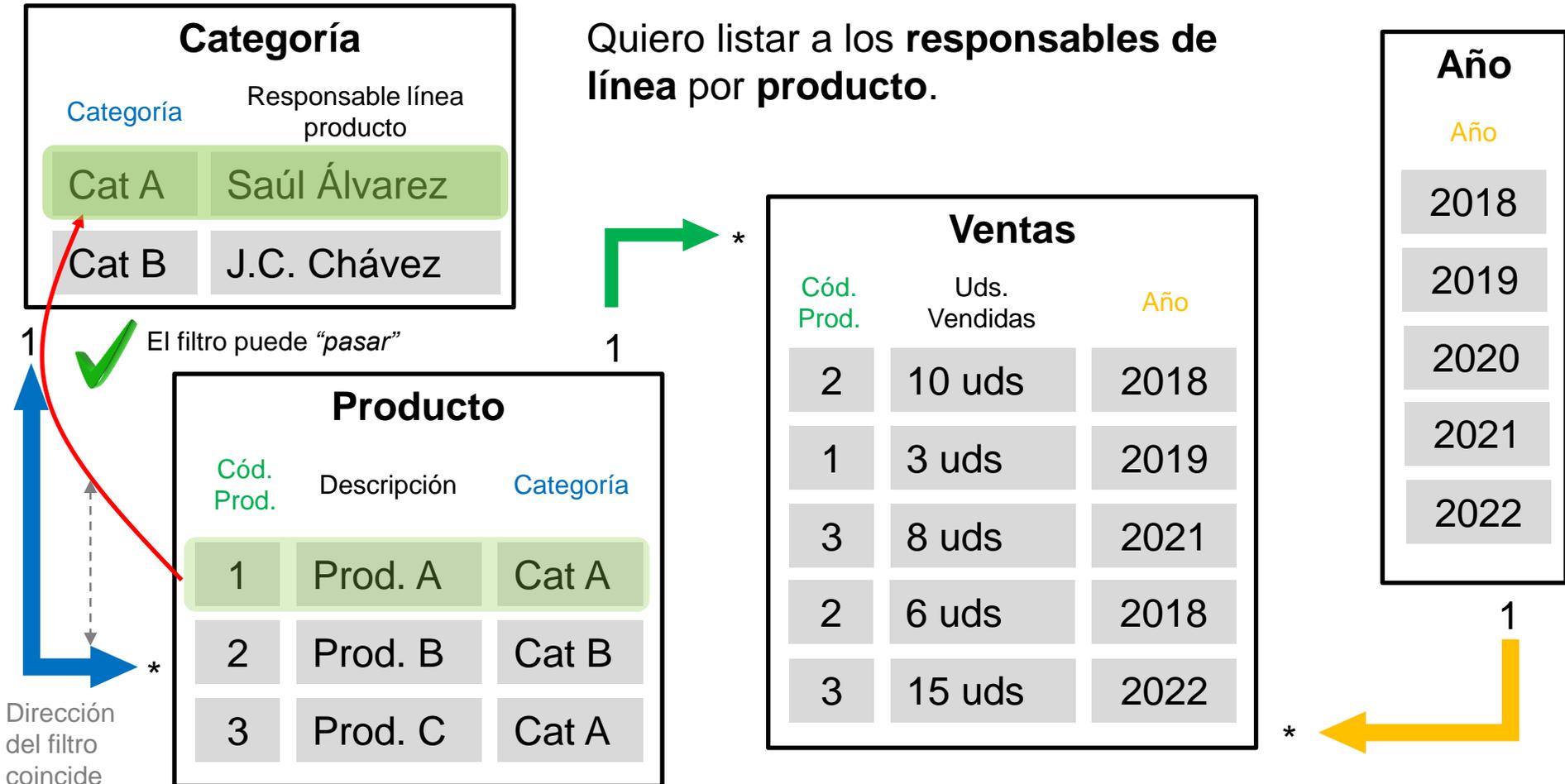
Ventas		
Cód. Prod.	Uds. Vendidas	Año
2	10 uds	2018
1	3 uds	2019
3	8 uds	2021
2	6 uds	2018
3	15 uds	2022



# Dirección del filtro cruzado



# Dirección del filtro cruzado



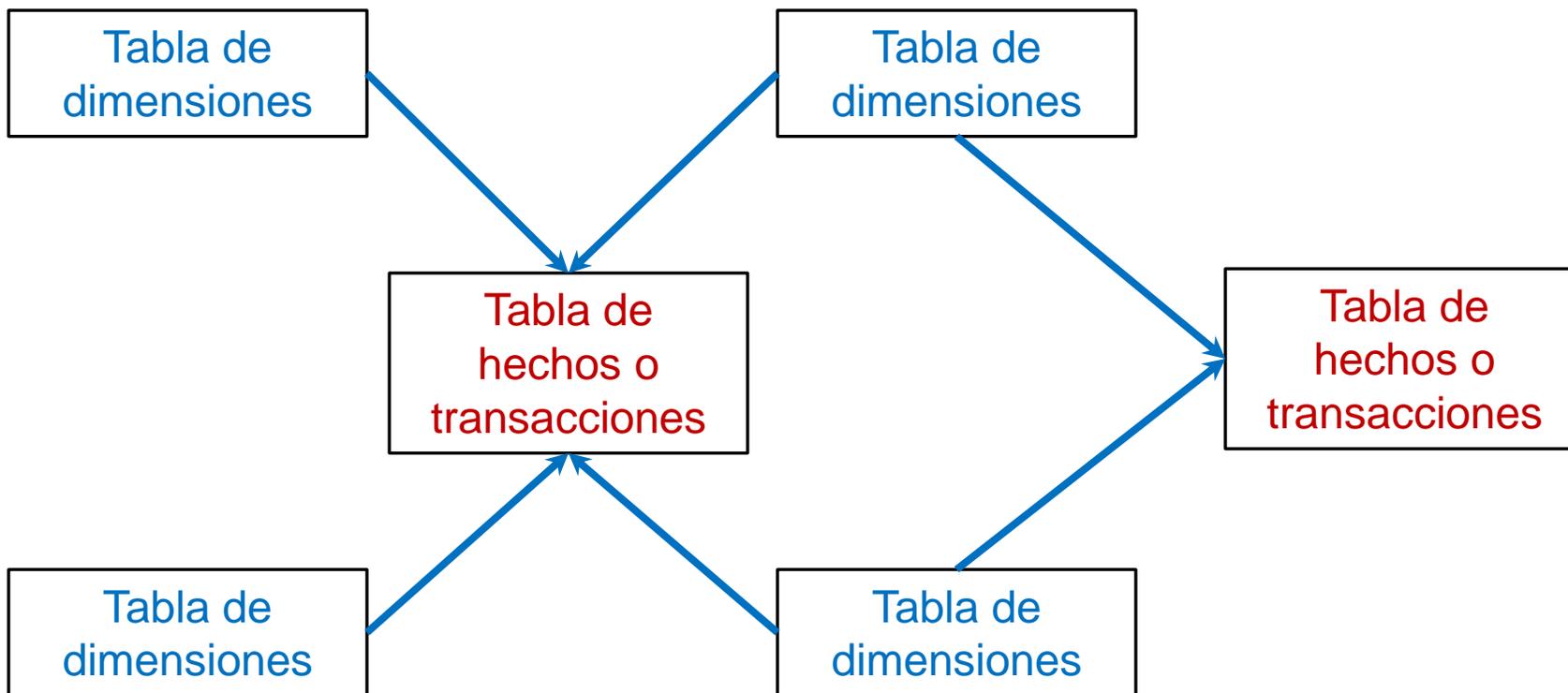
# Dirección el filtro cruzado

## Bidireccional (“Ambas”)



En un modelo de estrella, la dirección “Ambas” funciona porque el filtro puede ir de cualquier tabla de dimensiones a otra dado que hay un solo camino posible.

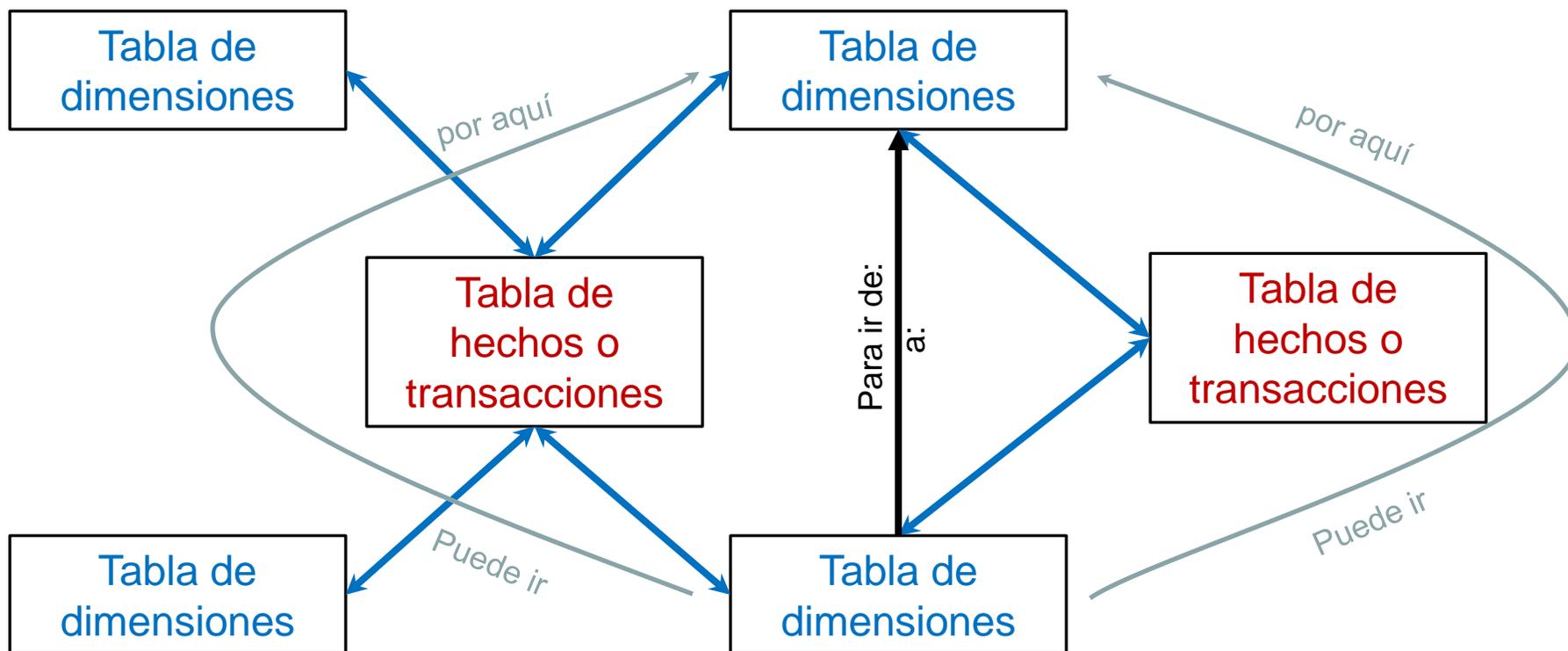
# Dirección el filtro cruzado



En un modelo de copo, la dirección “Ambas” no siempre funciona porque puede haber ambigüedad en el “camino” que el filtro tome para ir de una tabla a otra dado que puede que no haya un solo camino posible.

# Dirección el filtro cruzado

## Bidireccional (“Ambas”)



Existen varios caminos posibles para llevar un filtro de una tabla de dimensiones a otra.



# Contenidos

---

1. Modelo de datos
2. Transformación de datos
3. DAX
4. Modelo de datos – Ambigüedad

# Transformación de datos



# Organizar consultas en Power Query

Ejer Transformacion 1a - Grupos Power Query - Editor de Power Query

File Inicio Transformar Agregar columna Vista Herramientas Ayuda

Cerrar y aplicar Cerrar Nuevo origen Nueva consulta Orígenes recientes Especificar datos Configuración de origen de datos Administrar parámetros Parámetros Actualizar vista previa Consultas

Consultas [4]

- Importar [4]
- NomClient y Comer...
- Facturacion
- Series facturacion
- Nombres clientes
- Otras consultas

	A <sup>B</sup> C Nombre Cliente	A <sup>B</sup> C Comercial
1	Los lunes	Javier Ruiz
2	Los martes	Amaia Gil
3	Pérez escayolistas	Amaia Gil
4	Arte gráfico	Amaia Gil
5	The lucky guys	Javier Ruiz
6	GHI Este	Francisco Cervantes

Propiedades del grupo

Nombre: Combinación

Descripción: Contiene la tabla de dimensiones combinada para confeccionar un modelo de datos tipo "Estrella".

Aceptar Cancelar

Ejer Transformacion 1a - Grupos Power Query - Editor de Power Query

File Inicio Transformar Agregar columna Vista Herramientas

Cerrar y aplicar Cerrar Nuevo origen Nueva consulta Orígenes recientes Especificar datos Configuración de origen de datos Administrar parámetros Parámetros

Consultas [4]

- Importar [4]
- NomClient y Comer...
- Facturacion
- Series facturacion
- Nombres clientes
- Combinación
- Otras consultas

Contiene la tabla de dimensiones combinada para confeccionar un modelo de datos tipo "Estrella".



# Organizar consultas en Power Query

Ejer Transformacion 1a - Grupos Power Query - Editor de Power Query

File Inicio Transformar Agregar columna Vista Herramientas Ayuda

Cerrar y aplicar Cerrar Nuevo origen Nueva consulta Orígenes recientes Especificar datos Configuración de origen de datos Orígenes de datos Administrar parámetros Parámetros Actualizar vista previa Consultas Propiedades Editor avanzado Administrar

Consultas [4]

- Importar [4]
- NomClient y Comer...
- Facturacion
- Series facturacion
- Nombres clientes
- Otras consultas

= Table.TransformColumnTypes("#Encabezados

	A <sup>B</sup> C Nombre Cliente	A <sup>B</sup> C Comercial
1	Los lunes	Javier Ruiz
2	Los martes	Amaia Gil
3	Pérez escayolistas	Amaia Gil
4	Arte gráfico	Amaia Gil
5	The lucky guys	Javier Ruiz
6	GHI Este	Francisco Cervantes

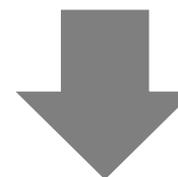
Pegar  
Nueva consulta  
Parámetro nuevo...  
Nuevo grupo...  
Expandir todo  
Contraer todo

Propiedades del grupo

Nombre  
Combinación

Descripción  
Contiene la tabla de dimensiones combinada para confeccionar un modelo de datos tipo "Estrella".

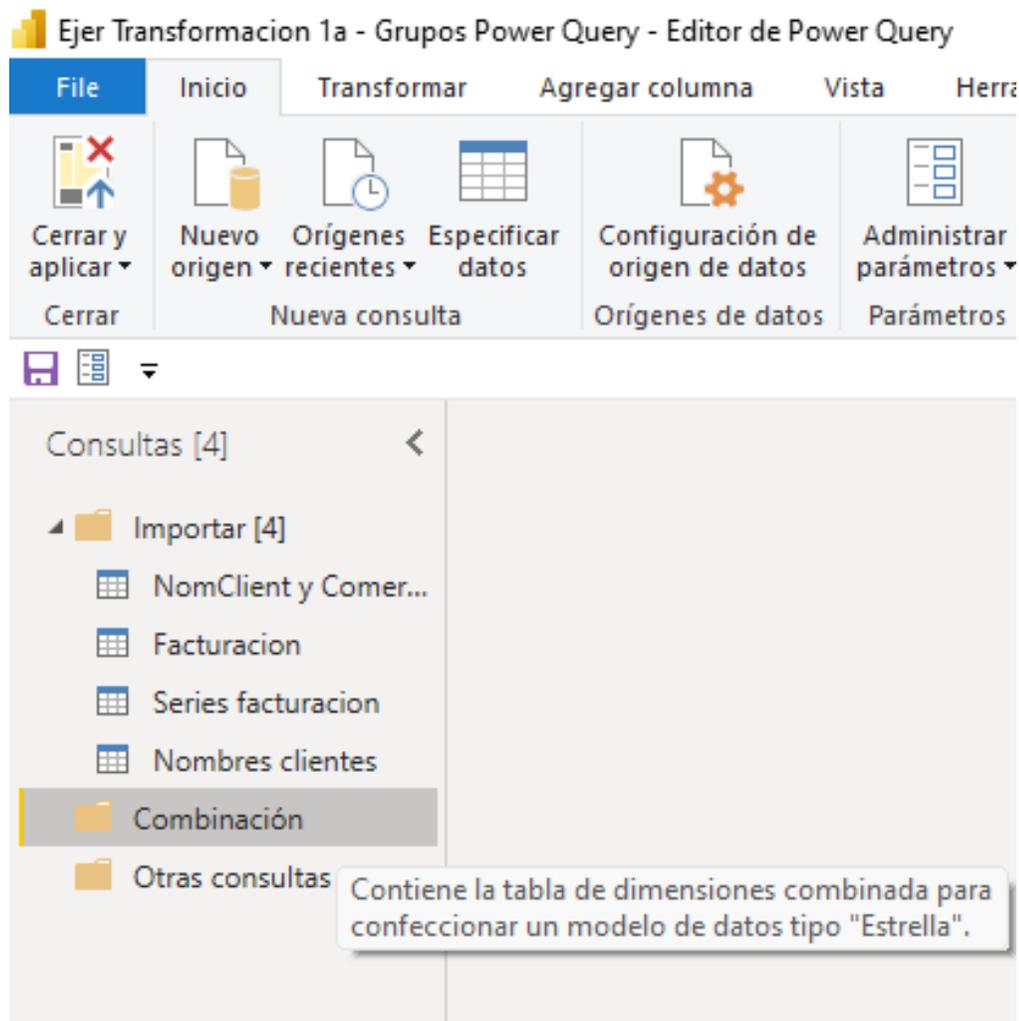
Aceptar Cancelar



Esto nos lleva a...

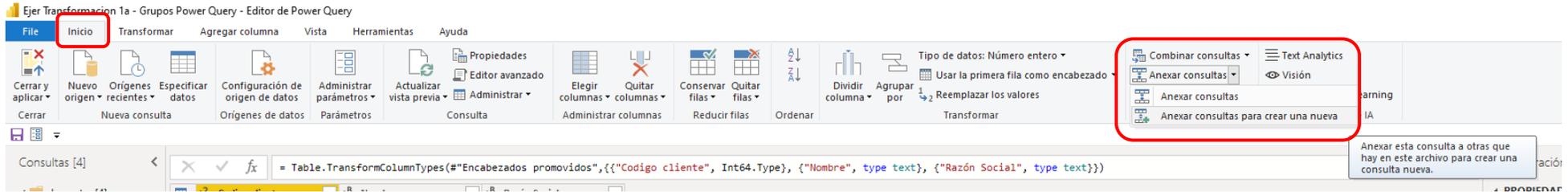
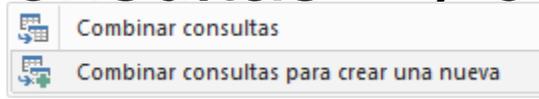
**FRANCISCO CERVANTES**  
Consultor Financiero

# Organizar consultas en Power Query

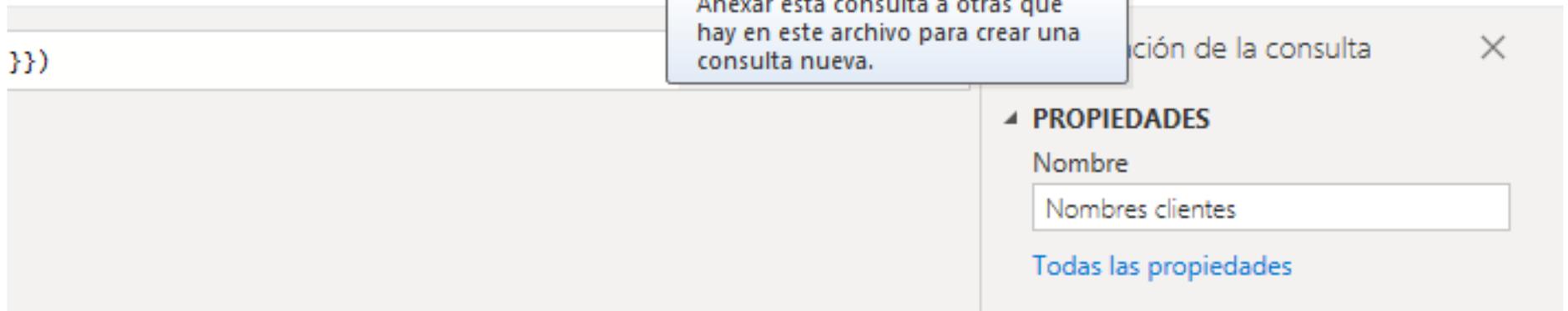


En esta recién creada carpeta podremos archivar otras consultas.

# fc Combinar consultas – y con ello, tablas – en Power Query



Anexar esta consulta a otras que hay en este archivo para crear una consulta nueva.



# fc Combinar consultas – y con ello, tablas – en Power Query

Combinar

Seleccione tablas y columnas coincidentes para crear una tabla combinada.

Nombres clientes

Codigo cliente	Nombre	Razón Social
1	Los lunes	ABC, S.L.
2	Los martes	DEF, S.A.
3	Pérez escayolistas	Juan Pérez autónomo
4	Arte gráfico	Ars, S.L.U
5	The lucky guys	Luk Co. Ltd.

NomClient y Comercial

Nombre Cliente	Comercial
Los lunes	Javier Ruiz
Los martes	Amaia Gil
Pérez escayolistas	Amaia Gil
Arte gráfico	Amaia Gil
The lucky guys	Javier Ruiz

Tipo de combinación

Externa izquierda (todas de la primera, coincidencias...)

Use las coincidencias aproximadas para comparar la combinación.

► Opciones de coincidencia aproximada

✓ La selección coincide con 5 de 5 filas de la primera tabla.

Aceptar Cancelar

# fc Combinar consultas – y con ello, tablas – en Power Query

Consultas [5]

Importar [5]

- NomClient y Comer...
- Facturacion
- Series facturacion
- Nombres clientes
- Combinar1
- Combinación
- Otras consultas

Formula Bar: = Table.NestedJoin(#"Nombres clientes", {"Nombre"}, #"NomClient y Comercial", {"Nombre Cl...

	1 <sup>2</sup> <sub>3</sub> Codigo cliente	A <sup>B</sup> <sub>C</sub> Nombre	A <sup>B</sup> <sub>C</sub> Razón Social	NomClient y Comercial
1		1 Los lunes	ABC, S.L.	Table
2		2 Los martes	DEF, S.A.	Table
3		3 Pérez escayolistas	Juan Pérez autónomo	Table
4		4 Arte gráfico	Ars, S.L.U	Table
5		5 The lucky guys	Luk Co. Ltd.	Table

Mover al grupo 'Combinación'

Configuración de la consulta

PROPIEDADES

Nombre

Cte Nombre RazSoc y Comercial

Todas las propiedades

# fc Combinar consultas – y con ello, tablas – en Power Query

The screenshot shows the Power Query interface. On the left, the 'Consultas [5]' pane lists 'Importar [4]' and 'Combinación [1]'. The main area displays a table with columns: 'Codigo cliente', 'Nombre', 'Razón Social', and 'NomClient y Comercial'. The formula bar shows: `= Table.NestedJoin("#Nombres clientes", {"Nombre"}, "#NomClient y Comercial", {"Nombre Cl...`. A context menu is open over the 'NomClient y Comercial' column, with 'Cambiar nombre' selected. An 'Expand' dialog box is also open, showing options to expand the column, with 'Expandir' selected and 'Usar el nombre de columna original como prefijo' checked.

	Codigo cliente	Nombre	Razón Social	NomClient y Comercial
1		Los lunes	ABC, S.L.	Table
2		Los martes	DEF, S.A.	Table
3		Pérez escayolistas	Juan Pérez autónomo	Table
4		Arte gráfico	Ars, S.L.U	Table
5		The lucky guys	Luk Co. Ltd.	Table

Context menu options:

- Copiar
- Pegar
- Eliminar
- Cambiar nombre
- Habilitar carga
- Incluir en la actualización del informe
- Duplicar
- Referencia
- Mover a un grupo
- Subir
- Bajar
- Crear función...
- Convertir en parámetro
- Editor avanzado
- Propiedades...

Expand dialog options:

- Expandir (selected)
- Agregar
- (Seleccionar todas las columnas)
- Nombre Cliente
- Comercial
- Usar el nombre de columna original como prefijo

# fc Combinar consultas – y con ello, tablas – en Power Query

Ejer Transformacion 1a - Grupos Power Query REH - Editor de Power Query

File Inicio Transformar Agregar columna Vista Herramientas Ayuda

Cerrar y aplicar Cerrar  
 Nuevo origen Nueva consulta  
 Orígenes recientes  
 Especificar datos  
 Configuración de origen de datos Orígenes de datos  
 Administrar parámetros Parámetros  
 Actualizar vista previa Consulta  
 Propiedades Editor avanzado Administrar  
 Elegir columnas Administrar columnas  
 Quitar columnas  
 Conservar filas Reducir filas  
 Quitar filas  
 Ordenar  
 Dividir columna Agrupar por Reemplazar los valores  
 Tipo de datos: Texto Usar la primera fila como encabezado Transformar

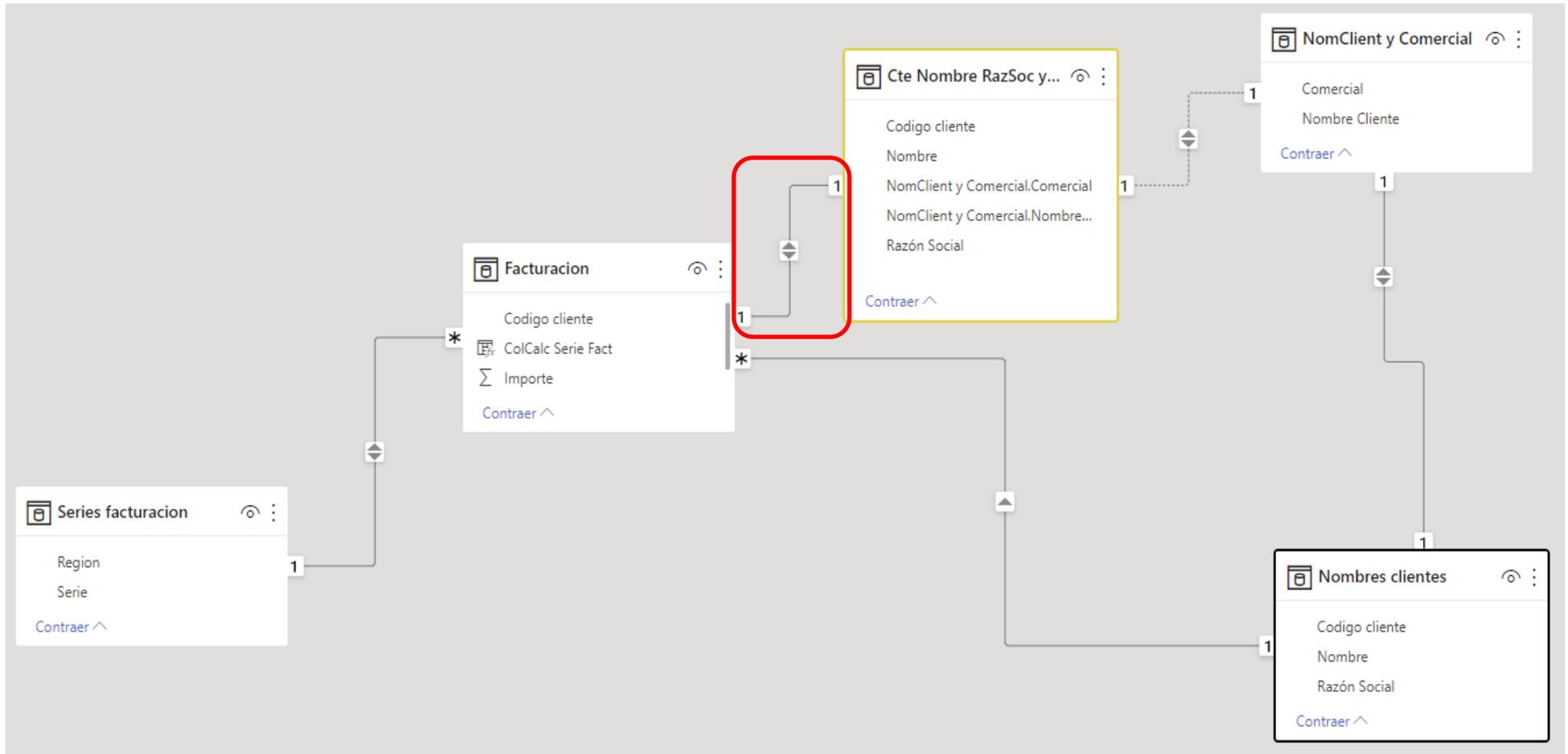
Consultas [5]

Importar [4]  
 NomClient y Comer...  
 Facturacion  
 Series facturacion  
 Nombres clientes  
 Combinación [1]  
 Cte Nombre RazSoc...  
 Otras consultas

= Table.ExpandTableColumn(Origen, "NomClient y Comercial", {"Nombre Cliente", "Comercial"}, {"NomClient y Comercial.Nombre Cliente", "NomC...})

1 <sup>2</sup> 3	Codigo cliente	A <sup>B</sup> C Nombre	A <sup>B</sup> C Razón Social	A <sup>B</sup> C NomClient y Comercial.Nombre Cliente	A <sup>B</sup> C NomClient y Comercial.Comercial
1	1	Los lunes	ABC, S.L.	Los lunes	Javier Ruiz
2	2	Los martes	DEF, S.A.	Los martes	Amaia Gil
3	3	Pérez escayolistas	Juan Pérez autónomo	Pérez escayolistas	Amaia Gil
4	4	Arte gráfico	Ars, S.L.U	Arte gráfico	Amaia Gil
5	5	The lucky guys	Luk Co. Ltd.	The lucky guys	Javier Ruiz

# fc Combinar consultas – y con ello, tablas – en Power Query



# fc Combinar consultas – y con ello, tablas – en Power Query

**Editar relación**

Permite seleccionar tablas y columnas relacionadas.

Facturacion

Codigo cliente	No. Factura	Importe	ColCalc Serie Fact
1	A-512	1.000,00 €	A
2	B-310	2.000,00 €	B
3	A-340	3.000,00 €	A

Cte Nombre RazSoc y Comercial

Codigo cliente	Nombre	Razón Social	NomClient y Comercial.Nombre Cliente	NomClient y Comercia
1	Los lunes	ABC, S.L.	Los lunes	Javier Ruiz
2	Los martes	DEF, S.A.	Los martes	Amaia Gil
3	Pérez escayolistas	Juan Pérez autónomo	Pérez escayolistas	Amaia Gil

Cardinalidad: Varios a uno (\*:1)

Dirección del filtro cruzado: Ambas

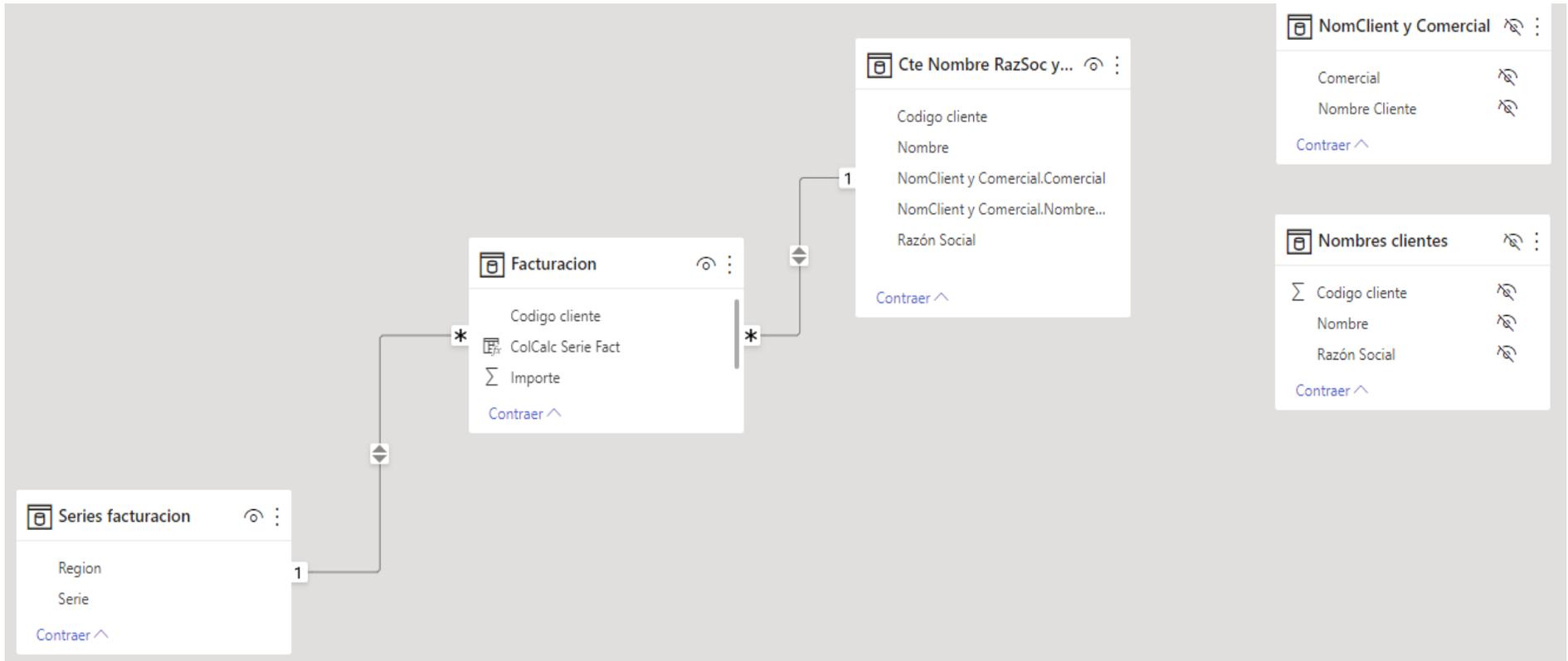
Activar esta relación

Aplicar filtro de seguridad en ambas direcciones

Asumir integridad referencial

Aceptar Cancelar

# fc Combinar consultas – y con ello, tablas – en Power Query



# fc Combinar consultas – y con ello, tablas – en Power Query

The screenshot displays the Power Query interface. On the left, two visualization placeholders are shown with error messages: "No se puede mostrar el objeto visual. Ver detalles" and a "Corregir esto" button. On the right, three data tables are visible:

Codigo cliente	Primera fecha: Region
1	Este
2	Este
3	Este
4	Este
5	Este

Codigo cliente	Primera fecha: Region
1	Sur
2	Norte
3	Sur
4	Sur
5	Norte

Codigo cliente	Primera fecha: Razón So
1	ABC, S.L.
2	ABC, S.L.
3	ABC, S.L.
4	ABC, S.L.
5	ABC, S.L.

On the far right, the 'Visualizaciones' and 'Campos' panes are visible. The 'Visualizaciones' pane shows filter settings for 'Nombre' (Comercial) and 'Region' (Region), with 'Valores' set to 'Suma de Importe'. The 'Campos' pane shows a list of fields with checkboxes, including 'Cte Nombre RazSoc y ...', 'Codigo cliente', 'Nombre', 'NomClient y Co...', 'NomClient y Co...', 'Razón Social', 'Facturacion', 'Codigo cliente', 'ColCalc: Serie Fact', 'Importe', 'No. Factura', 'Series facturacion', 'Region', and 'Serie'.

# fc Combinar consultas – y con ello, tablas – en Power Query

The image displays two screenshots of the Power Query interface, illustrating the process of combining queries and applying filters. The interface is divided into two main panes: 'Visualizaciones' (Visualizations) and 'Campos' (Fields).

**Left Screenshot (Initial State):**

- Visualizaciones:** Shows a grid of visualization options. The 'Filas' (Rows) section contains 'Nombre' and 'Comercial'. The 'Columnas' (Columns) section contains 'Region'. The 'Valores' (Values) section contains 'Suma de Importe'.
- Campos:** Shows a search bar and a list of fields. Under 'Cte Nombre RazSoc y ...', 'Codigo cliente', 'Nombre', 'NomClient y Co...', and 'Razón Social' are listed. Under 'Facturacion', 'Codigo cliente', 'ColCalc Serie Fact', 'Importe' (checked), and 'No. Factura' are listed. Under 'Series facturacion', 'Region' (checked) and 'Serie' are listed.

**Right Screenshot (Filtered State):**

- Visualizaciones:** The same grid of visualization options is shown, with the same row and column selections as the left screenshot.
- Campos:** The same search bar and list of fields are shown, but with additional filters applied. Under 'Cte Nombre RazSoc y ...', 'Comercial' and 'Nombre' are checked. Under 'Facturacion', 'Importe' and 'No. Factura' are checked. Under 'Series facturacion', 'Region' is checked.

An arrow points from the left screenshot to the right screenshot, indicating the transition from the initial state to the filtered state.



# Contenidos

---

1. Modelo de datos
2. Transformación de datos
3. DAX
4. Modelo de datos – Ambigüedad

# DAX

---

# Vista de consultas DAX



# Vista de consultas DAX – Interfaz

The screenshot displays the Power BI Desktop application window. At the top, the title bar reads "Sin título - Power BI Desktop" and includes window control buttons and a "Iniciar sesión" button. The ribbon is set to "Inicio" (Home) and shows various editing and formatting options. The main workspace is divided into three sections: a top bar with "Ejecutar" (Execute) and "Update model with changes (0)" buttons; a central area for the DAX query, currently empty; and a bottom bar showing "Consulta 1" (Query 1) with a plus sign to add more queries. On the right side, the "Datos" (Data) pane is visible, showing "Tablas" (Tables) and "Modelo" (Model) tabs, with a search box labeled "Buscar" (Search). The bottom status bar indicates "1 consulta de 1" (1 query of 1) and a zoom level of "100 %".



## Comandos DAX (“Statements”)

---



**EVALUATE** → Evalúa una expresión (F5)



**VAR** → Define una variable



**RETURN** → Devuelve el resultado de una variable o de un código DAX.

# Comandos DAX (“Statements”)

---



**ORDER BY** → Ordena una tabla



**DEFINE** → Comando que “alerta” a Power Bi de que vamos a introducir una medida para usar dentro de un código DAX o para mostrar su resultado.



**MEASURE** → Comando que precede al nombre de la medida a crear en el código DAX para usar dentro de un código o mostrar su resultado.

# Comandos DEFINE y MEASURE – Ejemplo

## DEFINE

Actualizar modelo: agragar nueva medida

```
MEASURE M_BASE[ImporteVentas]=  
    SUMX(TVentas,  
        TVentas[Unidades]*TVentas[Precio_venta_unitario])
```

Si definimos una medida en la Vista de Consultas DAX y dicha medida no existe en nuestro modelo, podemos agregar dicha medida a nuestro modelo desde la Vista de Consultas DAX.

---

# Funciones DAX

## Filtro y manipulación de tablas



# Función FILTER

---

**FILTER**(<table>,<filter>)

**FILTER** (Tabla, Filtro)

Devuelve la parte de una tabla según el criterio que apliquemos en el argumento “Filtro”.

## Notas:

- El criterio por el que se filtra la tabla se ha de encontrar dentro de la misma tabla.
- La tabla puede ser una tabla existente o una expresión que devuelva una tabla.



# Función FILTER

EVALUATE

`FILTER(ComboCteVendedor, [Zona]="Europa")`

Tabla a resumir

Filtro aplicado

	ComboCteVendedor[Co...	ComboCteVendedor[Zo...	ComboCteVendedor[Ci...	ComboCteVendedor[Pais]	ComboCteVendedor[Cli...
1	1	Europa	Aachen	Alemania	Drachenblut Delikatessen
2	1	Europa	Berlín	Alemania	Alfkiii
3	1	Europa	Brandenburg	Alemania	Königlich Essen
4	1	Europa	Cunewalde	Alemania	QUICK-Stop
5	1	Europa	Frankfurt a.M.	Alemania	Lehmans Marktstand
6	1	Europa	Köln	Alemania	Ottilies Käseladen
7	1	Europa	Leipzig	Alemania	Morgenstern Gesundkost
8	1	Europa	Mannheim	Alemania	Blauer See Delikatessen
9	1	Europa	München	Alemania	Frankenversand
10	1	Europa	Münster	Alemania	Toms Spezialitäten
11	1	Europa	Stuttgart	Alemania	Die Wandernde Kuh
12	1	Europa	Graz	Austria	Ernst Handel
13	1	Europa	Salzburg	Austria	Piccolo und mehr
14	1	Europa	Bruxelles	Bélgica	Maison Dewey
15	1	Europa	Charleroi	Bélgica	Suprêmes délices
16	1	Europa	Árhus	Dinamarca	Vaffeljernet
17	1	Europa	København	Dinamarca	Simons bistro
18	1	Europa	Barcelona	España	Galería del gastrónomo
19	1	Europa	Madrid	España	Bólido Comidas prepara...
20	1	Europa	Madrid	España	Romero y tomillo

Muestra solamente la parte de la tabla “ComboCteVendedor” correspondiente a la zona “Europa”.



# Función FILTER aplicada

EVALUATE

```
    ADDCOLUMNS(  
        CALCULATETABLE(  
            TVentas,  
            FILTER(ComboCteVendedor,[Zona] = "Europa")  
        ),  
        "ImporteTotalVtasCalc", [ImporteVentas],  
        "Diferencia Vtas", [ImporteVentas] - TVentas[Ventas_Brutas]  
    )
```

Devuelve:

- la parte de la **tabla TVentas** correspondiente sólo a las ventas de la zona “Europa” (**CALCULATETABLE + FILTER**)
- añadiendo a esta tabla las columnas “ImporteTotalVtasCalc” y “Diferencia Vtas” (**ADDCOLUMNS**).



# Función ADDCOLUMNS

---

## ADDCOLUMNS

(<table>, <name>, <expression>[, <name>, <expression>]...)

## ADDCOLUMNS

(Tabla, Nombre, Expresión [, Nombre, Expresión]...)

Devuelve:

- Una tabla original (Tabla) con columnas añadidas cuyo(s) nombre(s) es/son los definidos en el/los argumento(s) Nombre (escritos entre comillas) y calculados según el cálculo definido en el/los argumento(s) Expresión.



# Función CALCULATETABLE

---

## CALCULATETABLE

(<expression>[, <filter1> [, <filter2> [, ...]])

## CALCULATETABLE

(Expresión o Tabla, Filtro 1 [, Filtro 2, Filtro 3]...)

Devuelve:

- Una tabla filtrada según los filtros especificados en los argumentos Filtro 1, Filtro 2, etc.



# Función SUMMARIZE

---

**SUMMARIZE** (<table>, <groupBy\_columnName>[, <groupBy\_columnName>]...[, <name>, <expression>]...)

**SUMMARIZE** (Tabla, Agrupar por [, Agrupar por]...[, Nombre] [, Expresión]...)

Devuelve:

- Una tabla de valores únicos correspondientes a la columna seleccionada en el argumento “Agrupar por”.
- Una tabla de valores y totales agrupados según la columna seleccionada en el argumento “Agrupar por”.



# Función SUMMARIZE

---

**SUMMARIZE** (Tabla, Agrupar por [, Agrupar por]...[, Nombre] [, Expresión]...)

Tabla:

- Nombre de una tabla existente.
- Una *expresión* (cálculo) que dé como resultado – es decir, que *devuelva* – una tabla.

Agrupar por:

- Categoría (es decir, columna) cuyos valores únicos queremos mostrar o bien, categoría por la que deseamos agrupar los valores de la tabla; en caso de que haya valores repetidos, nos devolverá una lista de valores únicos.

Puede ser una columna dentro de la tabla que estamos sintetizando o bien la columna de una tabla relacionada.

Nombre:

- Texto entre comillas correspondiente al nombre de la columna que deseemos añadir. Podemos añadir varias columnas.

Expresión:

- *Cálculo* para obtener los datos de la(s) columna(s) que deseemos añadir. Esta *expresión* ha de dar como resultado un *valor escalar* (un texto o un número).



# Función SUMMARIZE

## EVALUATE

```
SUMMARIZE(TVentas, TVentas[Cliente_nombre])
```

Tabla a resumir

Columna dentro de la misma tabla cuyos valores únicos queremos mostrar.

Resultados

Resultado 1 de 1

	TVentas[Cliente_nombre]
1	Ottilies Käseladen
2	Frankenversand
3	Toms Spezialitäten
4	Ernst Handel
5	Suprêmes délices
6	Hanari Carnes
7	Wellington Importadora
8	Que Delícia

82	Let's Stop N Shop
83	Wilman Kala
84	Alfkiii
85	The Cracker Box
86	France restauration
87	Spécialités du monde
88	Maison Dewey
89	La corne d'abondance

Muestra los clientes (*sin repetir*) de la tabla TVentas.



# Función SUMMARIZE

## EVALUATE

```
SUMMARIZE(TVentas, TVentas[Cliente_nombre], TVentas[Producto])
```

Tabla a resumir

Columnas dentro de la misma tabla cuyos valores queremos mostrar.

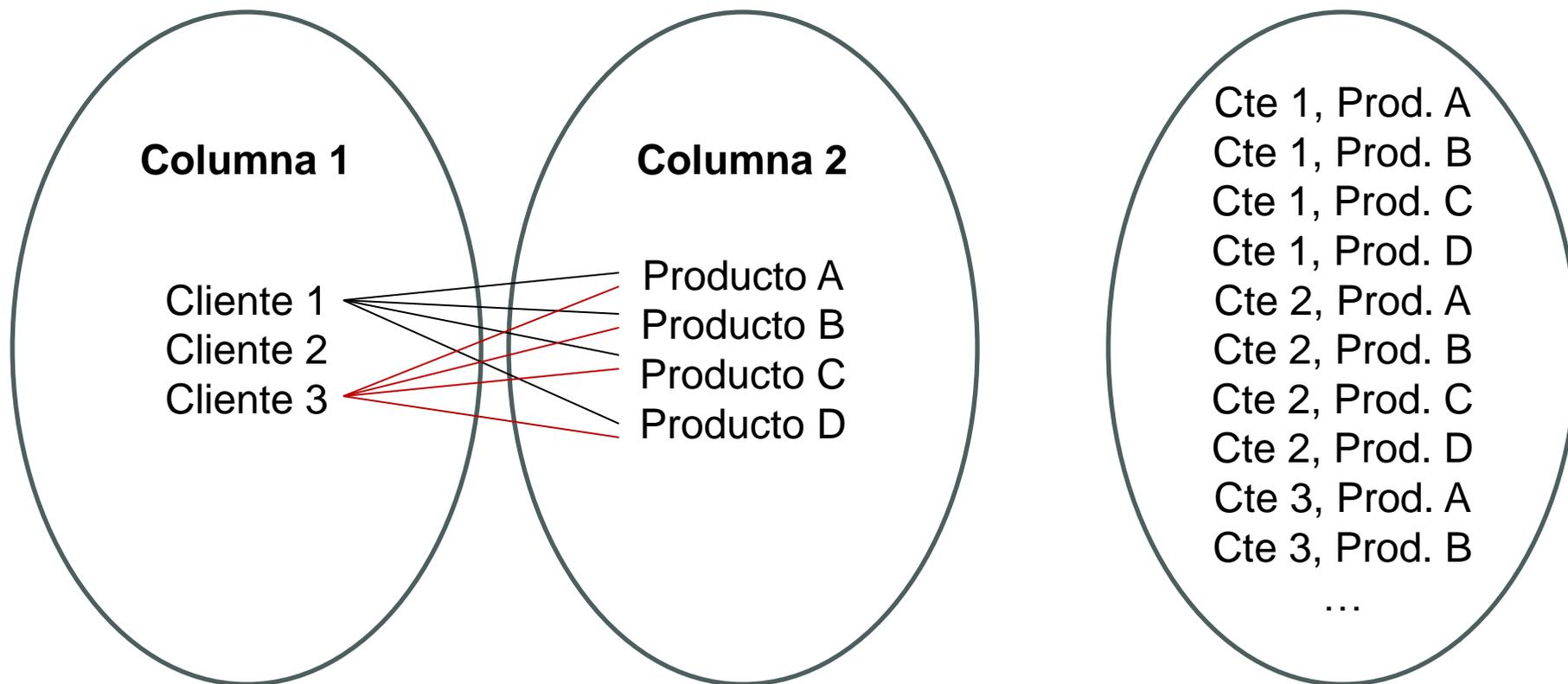
Resultados | Resultado 1 de 1 v Copia v

	TVentas[Cliente_nombre]	TVentas[Producto]
1	Otilies Käseladen	Raviolis Angelo
2	Otilies Käseladen	Cuajada de judías
3	Otilies Käseladen	Manzanas secas Manjim...
4	Otilies Käseladen	Crema de almejas estilo...
5	Otilies Käseladen	Carne de cangrejo de B...
6	Otilies Käseladen	Tarta de azúcar
7	Otilies Käseladen	Cerveza Outback
8	Otilies Käseladen	Postre de merengue Pa...

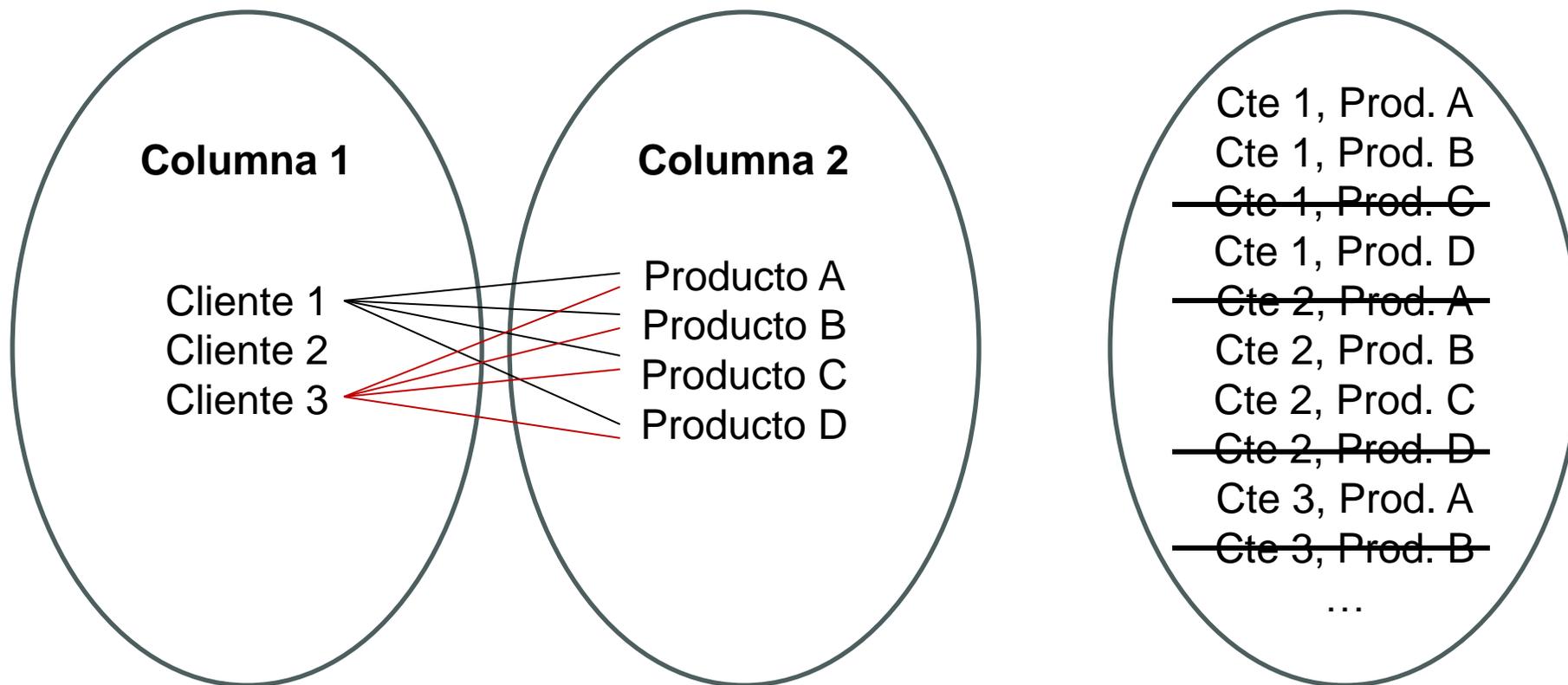
1606	La corne d'abondance	Queso de cabra
1607	La corne d'abondance	Mermelada de Sir Rodn...
1608	La corne d'abondance	Peras secas orgánicas d...
1609	La corne d'abondance	Cordero Alice Springs
1610	La corne d'abondance	Chocolate Schoggi
1611	La corne d'abondance	Cerveza Klosterbier Rhö...
1612	La corne d'abondance	Ositos de goma Gumbär
1613	La corne d'abondance	Cereales para Filo

Muestra los productos vendidos a cada cliente.

# Función SUMMARIZE



# Función SUMMARIZE



**Tabla  
resumen**



# Función SUMMARIZE

## EVALUATE

```
SUMMARIZE(TVentas, TVentas[Cliente_nombre],  
          TVentas[Producto], TVentas[Unidades])
```

Tabla a resumir

Columnas dentro de la misma tabla cuyos valores únicos queremos mostrar.

	TVentas[Cliente_nombre]	TVentas[Producto]	TVentas[Unidades]
1	Ernst Handel	Mezcla Gumbo del chef...	65
2	Ernst Handel	Arenque blanco del nor...	60
3	Ernst Handel	Cerveza tibetana Barley	50
4	Wartian Herkku	Queso Manchego La Pa...	12
5	Victuailles en stock	Pan de centeno crujient...	6
6	Victuailles en stock	Raviolis Angelo	15
7	Victuailles en stock	Salsa de pimiento pican...	20
8	Blondel père et fils	Cerveza Outback	

2184	White Clover Markets	Raclet de queso Courda...	30
2185	Save-a-lot Markets	Raclet de queso Courda...	100
2186	Save-a-lot Markets	Cerveza tibetana Barley	100
2187	White Clover Markets	Vino Côte de Blaye	25
2188	Save-a-lot Markets	Cordero Alice Springs	77
2189	Save-a-lot Markets	Salchicha Thüringer	60
2190	Richter Supermarkt	Empanada de carne	70
2191	Ernst Handel	Bollos de pan de Wimmer	130

Muestra las unidades de cada producto vendidas a cada cliente.



# Función SUMMARIZE

---

DEFINE

```
MEASURE M_BASE[ImporteVentas] =  
    SUMX(  
        TVentas,  
        TVentas[Unidades] * TVentas[Precio_venta_unitario]  
    )
```

EVALUATE

```
SUMMARIZE(  
    TVentas,  
    ComboCteVendedor[Zona],  
    ComboCteVendedor[País],  
    "Vtas", [ImporteVentas]  
)  
ORDER BY  
    ComboCteVendedor[Zona] DESC,  
    [Vtas] DESC
```



# Función SUMMARIZE

	ComboCteVendedor[Zo...	ComboCteVendedor[País]	[Vtas]
1	Europa	Alemania	299905.83
2	Europa	Austria	168351.02
3	Europa	Francia	104284.6
4	Europa	Reino Unido	93888.27
5	Europa	Suecia	80943.32
6	Europa	Irlanda	57699.79
7	Europa	Bélgica	41826.88
8	Europa	Dinamarca	34338.85

14	Europa	Noruega	6499.65
15	Europa	Polonia	4153.47
16	América del sur	Brasil	102979.96
17	América del sur	Venezuela	87486.89
18	América del sur	México	37035.9
19	América del sur	Argentina	13287.1
20	América del norte	Estados Unidos	280276.77
21	América del norte	Canadá	56317.9



# Función SUMMARIZECOLUMNS

---

## SUMMARIZECOLUMNS

( <groupBy\_columnName> [, <groupBy\_columnName >]..., [<filterTable>]...[, <name>, <expression>]...)

## SUMMARIZECOLUMNS

(Agrupar por [, Agrupar por] ...[, Tabla a filtrar]...[, Nombre] [, Expresión]...)

Devuelve el resumen de la Tabla a filtrar con valores agrupados según la(s) columna(s) seleccionada(s) en el/los argumentos “Agrupar por”.

Similar a la función SUMMARIZE, también podemos añadir columnas a la tabla resumen resultante.

Por razones técnicas y de rendimiento, se aconseja usar la función **SUMMARIZECOLUMNS** en lugar de la función SUMMARIZE.



# SUMMARIZE vs. SUMMARIZECOLUMNS

---

## SUMMARIZE

(Tabla, Agrupar por [, Agrupar por]...[, Nombre] [, Expresión]...)

## SUMMARIZECOLUMNS

(Agrupar por [, Agrupar por] ...[, Tabla de filtro]...[, Nombre] [, Expresión]...)



# Función SUMMARIZECOLUMNS

---

## SUMMARIZECOLUMNS

(Agrupar por [, Agrupar por] ...[, Tabla de filtro]...[, Nombre] [, Expresión]...)

Agrupar por:

- Categoría (es decir, columna) cuyos valores únicos queremos mostrar o bien, categoría por la que deseamos agrupar los valores de la tabla; en caso de que haya valores repetidos, nos devolverá una lista de valores únicos.

Puede ser una columna dentro de la tabla que estamos sintetizando o bien la columna de una tabla relacionada.

Tabla de filtro:

- Nombre de una tabla existente de la que deseamos obtener su resumen.
- Una *expresión* (cálculo) que dé como resultado – es decir, que *devuelva* – una tabla.

Nombre:

- Texto entre comillas correspondiente al nombre de la columna que deseamos añadir. Podemos añadir varias columnas.

Expresión:

- *Cálculo* para obtener los datos de la(s) columna(s) que deseamos añadir. Esta *expresión* ha de dar como resultado un *valor escalar* (un texto o un número).

---

# Funciones DAX

## Tiempo e Inteligencia de tiempo



# Diferencia entre funciones de tiempo y de inteligencia de tiempo

---

## Tiempo

- Nos permiten realizar cálculos de tiempo:
  - Generar una tabla de fechas
  - A qué semana del año pertenece una fecha
  - Cuántos días/meses/años de diferencia hay entre dos fechas, etc.
- Devuelven una magnitud de tiempo:
  - Una columna de fechas
  - Una fecha o una hora
  - Un número de días, etc.

## Inteligencia de tiempo

- Nos permiten calcular magnitudes acumuladas y por ende, comparar datos en el tiempo:
  - Ventas de un trimestre
  - Beneficio del mes
- Devuelven una magnitud en el tiempo:
  - Importe vendido hasta la fecha
  - Unidades vendidas el año anterior
  - Diferencia en el coste de ventas entre dos ejercicios, etc.
- Para su buen funcionamiento es indispensable tener una tabla maestra de fechas.



# Generar una tabla maestra de fechas

---

## CALENDARAUTO ( [Mes fin año fiscal])

Mes fin año fiscal:

- Argumento **opcional**. Si se omite, Power BI tomará Diciembre – es decir, 12 – como el mes en el que finaliza el año fiscal.

## CALENDAR (Fecha inicio, Fecha fin)

Fecha inicio y Fecha fin:

- Tiene que ser una fecha:
  - Una fecha proveniente de una columna que Power Bi reconozca como tal
  - Una fecha calculada mediante la función **DATE**.

# La función CALENDARAUTO

Calendario Auto TEST = CALENDARAUTO(12) ó

Calendario Auto TEST = CALENDARAUTO()

	Date
	01/01/1916 0:00:00
	02/01/1916 0:00:00
	03/01/1916 0:00:00
	04/01/1916 0:00:00
	05/01/1916 0:00:00
	06/01/1916 0:00:00
	07/01/1916 0:00:00
	08/01/1916 0:00:00
	09/01/1916 0:00:00
	10/01/1916 0:00:00
	11/01/1916 0:00:00
	12/01/1916 0:00:00
	13/01/1916 0:00:00
	14/01/1916 0:00:00
	15/01/1916 0:00:00
	16/01/1916 0:00:00
	17/01/1916 0:00:00
	18/01/1916 0:00:00

¡Desde 1916!  
¿Por qué?





# La función CALENDARAUTO

IdCliente	Nombre	Apellido	Nacimiento	Género	Estado Civil	Educación	Ocupación
15903	Abigail	Barnes	domingo, 9 de julio de 1916	Orden ascendente			siona
21794	Alyssa	James	jueves, 15 de septiembre de 1916	Orden descendente			siona
22773	Autumn	Zhang	domingo, 23 de abril de 1916	Borrar orden			siona
27433	Barry	Garcia	domingo, 20 de agosto de 1916	Borrar filtro			siona
22769	Benjamin	Martin	sábado, 22 de octubre de 1916	Borrar todos los filtros			siona
12110	Bianca	Guo	viernes, 3 de febrero de 1916	Filtros de fecha			siona
24677	Blake	Lee	miércoles, 15 de febrero de 1916				siona
23727	Bruce	Serrano	jueves, 2 de marzo de 1916				siona
14406	Carrie	Schmidt	martes, 15 de agosto de 1916				siona
22766	Cassidy	Hayes	sábado, 5 de noviembre de 1916				siona
11214	Charles	Miller	lunes, 7 de noviembre de 1916				siona
27432	Craig	Ruiz	martes, 9 de mayo de 1916				siona
20534	Dalton	Lopez	lunes, 13 de marzo de 1916				siona
23726	Darrell	Xie	martes, 13 de junio de 1916				siona
29018	Drew	Luo	martes, 13 de junio de 1916				siona
14405	Dylan	Walker	jueves, 1 de junio de 1916				siona
17163	Edward	Adams	miércoles, 23 de agosto de 1916				siona
29007	Elijah	Perry	lunes, 26 de diciembre de 1916				siona
15335	Emily	Patterson	viernes, 3 de febrero de 1916				siona
11220	Erica	Huang	jueves, 23 de febrero de 1916				siona
27426	Evan	Stewart	jueves, 22 de diciembre de 1916				siona
12098	Grace	Clark	martes, 25 de octubre de 1916				siona
18093	Grace	Johnson	sábado, 5 de noviembre de 1916				siona
21482	Haley	Flores	sábado, 8 de octubre de 1916				siona
25887	Jack	Collins	miércoles, 24 de mayo de 1916				siona
17152	Jack	Wang	lunes, 7 de noviembre de 1916				siona
29016	Jackson	Perez	lunes, 20 de marzo de 1916				siona
12111	Jacqueline	Gonzales	viernes, 4 de agosto de 1916				siona
12108	James	Walker	jueves, 23 de febrero de 1916	Hombre	Casado/a	ERP	Profesiona

- (Seleccionar todo)
- miércoles, 5 de julio de 1916
- miércoles, 11 de octubre de 1916
- jueves, 7 de diciembre de 1916
- lunes, 15 de octubre de 1917
- sábado, 16 de febrero de 1918
- domingo, 19 de mayo de 1918
- martes, 8 de abril de 1919
- viernes, 6 de junio de 1919
- miércoles, 27 de agosto de 1919
- martes, 13 de julio de 1920
- jueves, 10 de marzo de 1921
- jueves, 10 de noviembre de 1921
- sábado, 4 de febrero de 1922
- lunes, 13 de febrero de 1922
- sábado, 3 de junio de 1922
- miércoles, 20 de septiembre de 1922
- miércoles, 11 de octubre de 1922
- miércoles, 23 de abril de 1924
- viernes, 20 de junio de 1924
- domingo, 6 de julio de 1924
- lunes, 18 de agosto de 1924
- lunes, 9 de marzo de 1925

Aceptar Cancelar

Porque la función CALENDARAUTO busca la fecha más antigua en **TODAS** las tablas del archivo y confecciona una tabla calendario desde la fecha más antigua que encuentre hasta la fecha más reciente.

En este caso, una de las tablas contiene fechas de nacimiento, la más antigua de las cuales es el 5 de julio de 1916.

Pero ese no es el conjunto de fechas que nos interesa para el análisis.



# Generar una tabla maestra de fechas

---

## CALENDARAUTO ( [Mes fin año fiscal])

Mes fin año fiscal:

- Argumento **opcional**. Si se omite, Power BI tomará Diciembre – es decir, 12 – como el mes en el que finaliza el año fiscal.

## CALENDAR (Fecha inicio, Fecha fin)

Fecha inicio y Fecha fin:

- Tiene que ser una fecha:
  - Una fecha proveniente de una columna que Power Bi reconozca como tal
  - Una fecha calculada mediante la función **DATE**.



# Función ADDCOLUMNS

---

**ADDCOLUMNS**(Tabla, Nombre ColCalc, Cálculo ColCalc [, Nombre, Cálculo]...)

Tabla:

- Nombre de una tabla existente.
- Una *expresión* (cálculo) que dé como resultado – es decir, que *devuelva* – una tabla.

Nombre ColCalc:

- Texto entre comillas correspondiente al nombre de la columna que deseemos añadir. Podemos añadir varias columnas.

Cálculo ColCalc:

- *Expresión* para calcular los datos de la(s) columna(s) que deseemos añadir. Esta *expresión* ha de dar como resultado un *valor escalar* (un texto o un número **únicos**).



# ADDCOLUMNS para crear tabla calendario

---

```
CALENDARIO =  
    ADDCOLUMNS(  
        CALENDAR(MIN(Ventas[Fecha]),MAX(Ventas[Fecha])),  
        "Año", YEAR([Date]),  
        "Trimestre", "T"&QUARTER([Date]),  
        "Mes", MONTH([Date]),  
        "MesNombre", FORMAT([Date],"mmm"),  
        "Día", DAY([Date]),  
        "Trim Fiscal", "T"&  
            IF(AND(MONTH([Date])>=9,MONTH([Date])<=11),1,  
            IF(OR(MONTH([Date])=12,MONTH([Date])<=2),2,  
            IF(AND(MONTH([Date])>=3,MONTH([Date])<=5),3,4)))  
    )
```



# ADDCOLUMNS para crear tabla calendario

```
CALENDARIO =  
    ADDCOLUMNS(  
        CALENDAR(MIN(Ventas[Fecha]),MAX(Ventas[Fecha])),
```



Tabla

```
        "Año", YEAR([Date]),  
        "Trimestre", "T"&QUARTER([Date]),  
        "Mes", MONTH([Date]),  
        "MesNombre", FORMAT([Date], "mmm"),  
        "Día", DAY([Date]),  
        "Trim Fiscal", "T"&  
            IF(AND(MONTH([Date])>=9,MONTH([Date])<=11),1,  
            IF(Tabla(OR(MONTH([Date])=12,MONTH([Date])<=2),2,  
            IF(AND(MONTH([Date])>=3,MONTH([Date])<=5),3,4)))  
    )
```

Nombre  
ColCalc

Cálc.  
ColCalc

# Tabla calendario como tabla de fechas

Archivo Inicio Ayuda **Herramientas de tablas**

Nombre Calendario

**Marcar como tabla de fechas** Administrar relaciones Nueva Medida rápida

Permite seleccionar una tabla para usarla en las fechas de este informe. Se hará referencia a esta tabla si se configuran las jerarquías de datos o la inteligencia de tiempo.

```

3 VAR FechaFinal= DATE(YEAR(MAX('Me
4 RETURN
5 CALENDAR(FechaInicial, FechaFinal)

```

Date	Trimestre	Mes Num	Mes
01/01/2019 0:00:00	1	1	ene
02/01/2019 0:00:00	1	1	ene
03/01/2019 0:00:00	1	1	ene
04/01/2019 0:00:00	1	1	ene
05/01/2019 0:00:00	1	1	ene

Es importante marcar nuestra recién creada tabla de fechas como tabla de fechas...

**Marcar como tabla de fechas**

Seleccione una columna para usarla para la fecha. La columna debe ser del tipo de datos "fecha" y contener únicamente valores únicos. [Más información](#)

Columna de fecha

Date

Seleccionar columna

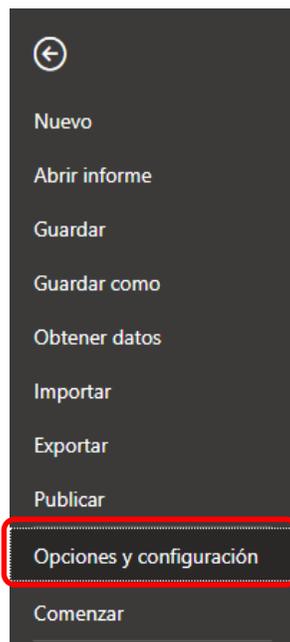
Date

...marán las tablas de fechas integradas asociadas con esta tabla. Es posible que los objetos visuales o las expresiones DAX que hagan referencia a ellas queden dañados. [Más información sobre cómo corregir los objetos visuales y las expresiones DAX](#)

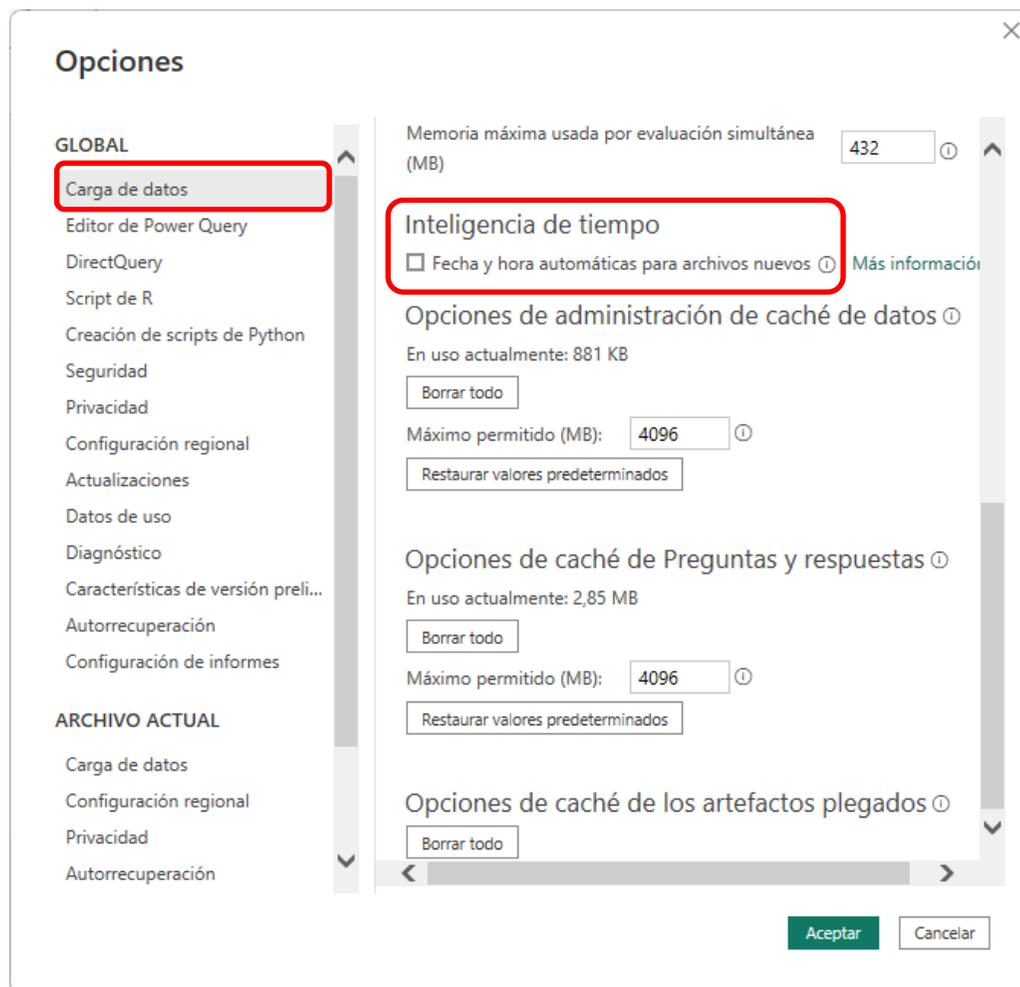
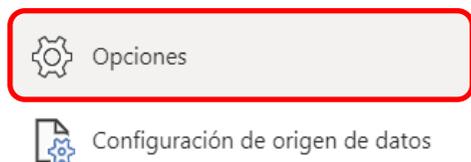
Aceptar Cancelar

# Tabla calendario como tabla de fechas

...y complementar este paso quitando la selección de “Fecha y hora automáticas” en Inteligencia de tiempo.



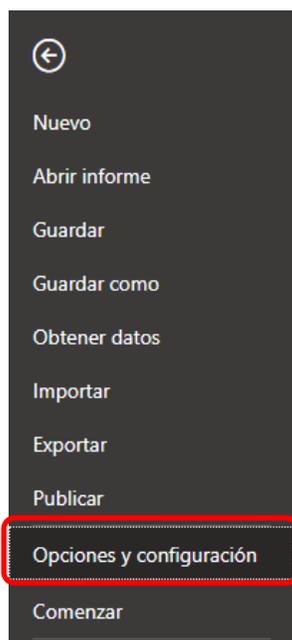
## Opciones y configuración



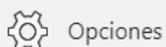


# Tabla calendario como tabla de fechas

Esto se puede hacer para todos los modelos de Power BI por defecto o para cada archivo creado.



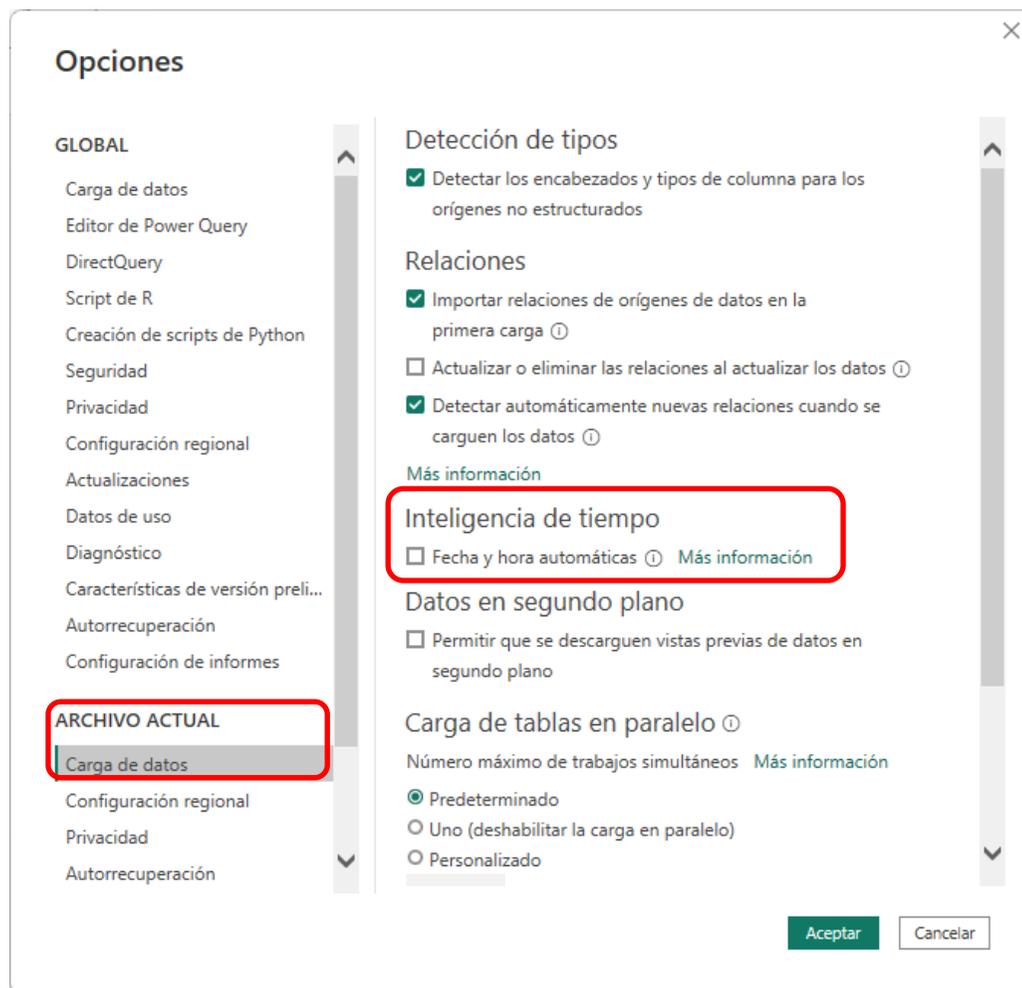
## Opciones y configuración



Opciones



Configuración de origen de datos





# Total acumulado hasta la fecha (Year-To-Date)

---

Usaremos las funciones **CALCULATE + DATESYTD**





# Función CALCULATE

---

Evalúa una expresión en un contexto de filtro modificado. Es decir, realiza un cálculo aplicando uno o varios filtros especificados **dentro** de la misma función.

**CALCULATE**(Expresión, [filtro<sub>1</sub>], [filtro<sub>1</sub>], ..., [filtro<sub>n</sub>])

Medida o cálculo a realizar  
(nunca una columna)

Contexto de filtro(s) en  
el(los) que ha de realizarse  
el cálculo (la expresión).

Es una de las 2 expresiones DAX capaz de **cambiar el contexto de filtro** dentro de la propia función (la otra es **CALCULATE**TABLE).



## Función DATESYTD

---

### DATESYTD (Fechas, [Fecha fin año])

Columna de fechas =  
Calendario[Date]

Fecha en que se termina el  
ejercicio fiscal. Si se omite  
= 31 de diciembre.

Devuelve una tabla con **una** columna de fechas que van desde el primer día del último año de la tabla Calendario hasta la última fecha que haya en dicha tabla.

El primer día del último año = al día siguiente del mes y día marcado en “Fecha fin año” correspondiente al último ejercicio para el que tenemos fechas.

Es decir:

- Para saber la **primera fecha** que nos devuelve la función DATESYTD nos hemos de hacer la siguiente pregunta: “¿Cuándo empezó el año fiscal en el que estamos?”
- Para saber la **última fecha** que nos devuelve la función DATES YTD, nos hemos de preguntar: ¿Cuál es la última fecha que tenemos en nuestra tabla calendario?”



# Función DATESYTD

Supongamos una tabla Calendario que va del 1-Abril-2020 al 25-Agosto-2024:

CALENDARIO = `CALENDAR`(`DATE`(2020,4,1),`DATE`(2024,8,25))

## Supuesto 1:

YTD = `DATESYTD`(CALENDARIO[Date], "2020-06-30")

Date
01/07/2024 0:00:00
02/07/2024 0:00:00
03/07/2024 0:00:00
04/07/2024 0:00:00

Fin año fiscal: **30-junio**

Según la fecha introducida en `DATESYTD`, el año fiscal empieza el 1 de julio.

20/08/2024 0:00:00
21/08/2024 0:00:00
22/08/2024 0:00:00
23/08/2024 0:00:00
24/08/2024 0:00:00
25/08/2024 0:00:00

El último 1-Julio, es decir, el 1-Julio más reciente que tenemos corresponde, en este caso, al **1-Julio-2024**.

## Supuesto 2:

YTD = `DATESYTD`(CALENDARIO[Date], "2020-08-26")

Date
27/08/2023 0:00:00
28/08/2023 0:00:00
29/08/2023 0:00:00
30/08/2023 0:00:00

Fin año fiscal: **26-agosto**

Según la fecha introducida en `DATESYTD`, el año fiscal empieza el 27 de agosto.

20/08/2024 0:00:00
21/08/2024 0:00:00
22/08/2024 0:00:00
23/08/2024 0:00:00
24/08/2024 0:00:00
25/08/2024 0:00:00

El último 27-Ago, es decir, el 27-Ago más reciente que tenemos corresponde, en este caso, al **27-Ago-2023**.

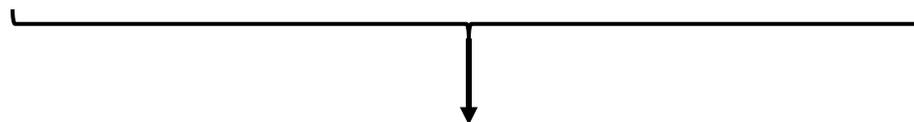


# Total Acumulado hasta la fecha (Year-To-Date)

**CALCULATE** ([Cantidad], **DATESYTD** (Calendar[Date]))

ó

**CALCULATE** ([Cantidad], **DATESYTD** (Calendar[Date], "12/31"))



Filtro de la función CALCULATE

Año	Cantidad	Cantidad YTD	Importe	Importe YTD
2018	25.140 uds	25.140 uds	605.417 €	605.417 €
2019	224.841 uds	224.841 uds	901.552 €	901.552 €
T1	6.946 uds	6.946 uds	122.736 €	122.736 €
T2	9.595 uds	16.541 uds	145.185 €	267.921 €
T3	86.960 uds	103.501 uds	265.386 €	533.307 €
T4	121.340 uds	224.841 uds	368.245 €	901.552 €
2020	360.679 uds	360.679 uds	2.192.134 €	2.192.134 €
T1	128.300 uds	128.300 uds	385.543 €	385.543 €
T2	159.814 uds	288.114 uds	512.405 €	897.948 €
T3	47.244 uds	335.358 uds	487.638 €	1.385.586 €
T4	25.321 uds	360.679 uds	806.548 €	2.192.134 €
<b>Total</b>	<b>610.660 uds</b>	<b>360.679 uds</b>	<b>3.699.103 €</b>	<b>2.192.134 €</b>



## Total del año anterior

---

Usaremos las funciones **CALCULATE + PREVIOUSYEAR**





# Función PREVIOUSYEAR

---

**PREVIOUSYEAR** (Fechas, [Fecha fin de año])

Columna de fechas =  
Calendario[Date]

Fecha en que se termina el  
ejercicio fiscal. Si se omite  
= 31 de diciembre.

Devuelve una tabla con **una** columna de fechas que correspondientes al año anterior de una fecha determinada.



# Función PREVIOUSYEAR

La siguiente consulta DAX devuelve una tabla vacía.

EVALUATE

PREVIOUSYEAR(CALENDARIO[Date])

Ejecutar Actualizar modelo con cambios (0)

```
1 EVALUATE
2 PREVIOUSYEAR(CALENDARIO[Date])
3
```

Resultados | Resultado 1 de 1 | Copia

CALENDARIO[Date]

**¿Por qué?**

Porque no hemos especificado el año previo con respecto a qué.



# Función PREVIOUSYEAR

La siguiente consulta DAX sí devuelve una columna de fechas:

EVALUATE

```
CALCULATETABLE(  
    PREVIOUSYEAR(CALENDARIO[Date]),  
    YEAR(CALENDARIO[Date]) = 2022  
)
```



Date
01/01/2021 0:00:00
02/01/2021 0:00:00
03/01/2021 0:00:00
04/01/2021 0:00:00
05/01/2021 0:00:00
06/01/2021 0:00:00
07/01/2021 0:00:00
08/01/2021 0:00:00
09/01/2021 0:00:00
10/01/2021 0:00:00
11/01/2021 0:00:00
12/01/2021 0:00:00
22/12/2021 0:00:00
23/12/2021 0:00:00
24/12/2021 0:00:00
25/12/2021 0:00:00
26/12/2021 0:00:00
27/12/2021 0:00:00
28/12/2021 0:00:00
29/12/2021 0:00:00
30/12/2021 0:00:00
31/12/2021 0:00:00

## ¿Por qué?

Porque estamos pidiendo que se calcule una tabla (**CALCULATETABLE**) con las fechas de la tabla Calendario correspondientes al año anterior (**PREVIOUSYEAR**) de un ejercicio concreto (el **2022**).



# Total del año anterior

## CALCULATE + PREVIOUSYEAR

Cant Año Pasado = CALCULATE(  
[Cantidad], PREVIOUSYEAR(CALENDARIO[Date])  
)



Año	Cantidad	Cantidad YTD	Cant Año Pasado	Importe	Importe YTD	Importe Año Pasado
2018	25.140 uds	25.140 uds		605.417 €	605.417 €	
2019	224.841 uds	224.841 uds	25.140 uds	901.552 €	901.552 €	605.417 €
T1	6.946 uds	6.946 uds	25.140 uds	122.736 €	122.736 €	605.417 €
T2	9.595 uds	16.541 uds	25.140 uds	145.185 €	267.921 €	605.417 €
T3	86.960 uds	103.501 uds	25.140 uds	265.386 €	533.307 €	605.417 €
T4	121.340 uds	224.841 uds	25.140 uds	368.245 €	901.552 €	605.417 €
2020	360.679 uds	360.679 uds	224.841 uds	2.192.134 €	2.192.134 €	901.552 €
T1	128.300 uds	128.300 uds	224.841 uds	385.543 €	385.543 €	901.552 €
T2	159.814 uds	288.114 uds	224.841 uds	512.405 €	897.948 €	901.552 €
T3	47.244 uds	335.358 uds	224.841 uds	487.638 €	1.385.586 €	901.552 €
T4	25.321 uds	360.679 uds	224.841 uds	806.548 €	2.192.134 €	901.552 €
<b>Total</b>	<b>610.660 uds</b>	<b>360.679 uds</b>		<b>3.699.103 €</b>	<b>2.192.134 €</b>	



# Comparativa: Acumulado hasta la fecha (Year-To-Date ó YTD) vs. Total año anterior

$$\text{Comp Uds YTD-FYAnt} = [\text{Cantidad YTD}] - [\text{Cant Año Pasado}]$$

Año	Cant Año Pasado	Cantidad YTD	Comp Uds YTD-FYAnt
2018		25.140 uds	25.140 uds
2019	25.140 uds	224.841 uds	199.701 uds
2020	224.841 uds	360.679 uds	135.838 uds
<b>Total</b>		<b>360.679 uds</b>	<b>360.679 uds</b>

Si preferimos indicar que no hay comparativa, entonces

$$\text{Comp Uds YTD-FYAnt} = \text{IF}(\text{NOT}(\text{ISBLANK}([\text{Cant Año Pasado}])), [\text{Cantidad YTD}] - [\text{Cant Año Pasado}])$$

Año	Cant Año Pasado	Cantidad YTD	Comp Uds YTD-FYAnt
2018		25.140 uds	
2019	25.140 uds	224.841 uds	199.701 uds
2020	224.841 uds	360.679 uds	135.838 uds
<b>Total</b>		<b>360.679 uds</b>	



## Total al mismo periodo del año anterior

---

Usaremos las funciones  
**CALCULATE + SAMEPERIODLASTYEAR**





# Función SAMEPERIODLASTYEAR

---

## SAMEPERIODLASTYEAR (Fechas)



Columna de fechas = Calendario[Date]

Devuelve una tabla con **una** columna de fechas que correspondientes al mismo periodo del año anterior con respecto a una fecha determinada.

# Función SAMEPERIODLASTYEAR

## EVALUATE

```

CALCULATETABLE(
  SAMEPERIODLASTYEAR(CALENDARIO[Date]),
  YEAR(CALENDARIO[Date]) = 2020,
  MONTH(CALENDARIO[Date]) = 4,
  DAY(CALENDARIO[Date]) >= 25
)

```

En este caso, buscamos el mismo periodo del año anterior con respecto al **25-Abril-2020**.

	CALENDARIO[Date]
1	4/25/2019 12:00:00 AM
2	4/26/2019 12:00:00 AM
3	4/27/2019 12:00:00 AM
4	4/28/2019 12:00:00 AM
5	4/29/2019 12:00:00 AM
6	4/30/2019 12:00:00 AM





# Total al mismo periodo del año anterior

Cant SAMEPERLY = `CALCULATE(`  
    `[Cantidad],`  
    `SAMEPERIODLASTYEAR(CALENDARIO[Date])`  
    `)`

Año	Cantidad	Cant SAMEPERLY
<b>2018</b>	<b>25.140 uds</b>	
T1	4.855 uds	
T2	6.428 uds	
T3	6.907 uds	
T4	6.950 uds	
<b>2019</b>	<b>224.841 uds</b>	<b>25.140 uds</b>
T1	6.946 uds	4.855 uds
T2	9.595 uds	6.428 uds
T3	86.960 uds	6.907 uds
T4	121.340 uds	6.950 uds
<b>2020</b>	<b>360.679 uds</b>	<b>224.841 uds</b>
T1	128.300 uds	6.946 uds
T2	159.814 uds	9.595 uds
T3	47.244 uds	86.960 uds
T4	25.321 uds	121.340 uds
<b>Total</b>	<b>610.660 uds</b>	<b>249.981 uds</b>





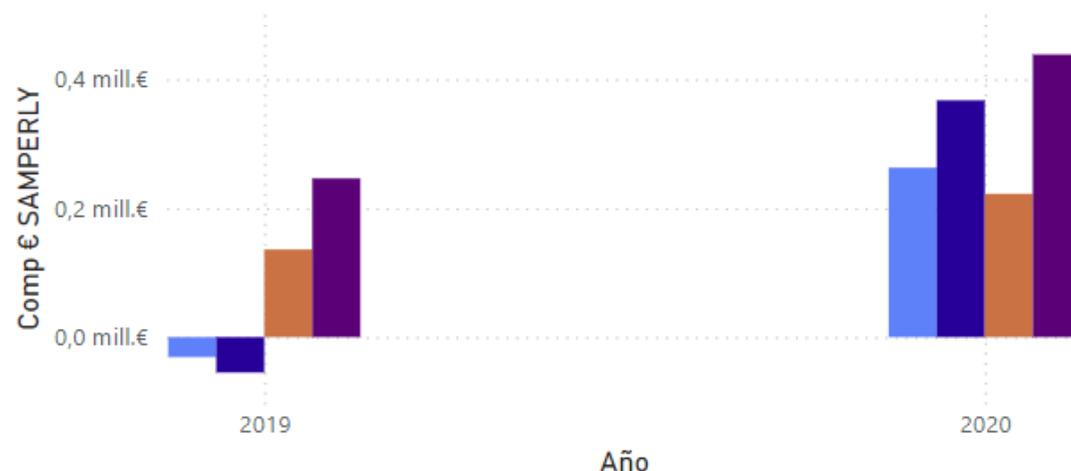
# Comparativa: Acumulado en periodo vs. acumulado en mismo periodo año anterior

Comp € SAMPERLY = IF(  
NOT(ISBLANK([Importe SAMPERLY])),  
[Importe]-[Importe SAMPERLY])

Año	Importe	Importe SAMPERLY	Comp € SAMPERLY
<b>2020</b>	<b>2.192.134 €</b>	<b>901.552 €</b>	<b>1.290.582 €</b>
T4	806.548 €	368.245 €	438.304 €
T3	487.638 €	265.386 €	222.252 €
T2	512.405 €	145.185 €	367.220 €
T1	385.543 €	122.736 €	262.807 €
<b>2019</b>	<b>901.552 €</b>	<b>605.417 €</b>	<b>296.134 €</b>
T4	368.245 €	122.144 €	246.101 €
T3	265.386 €	129.497 €	135.889 €
T2	145.185 €	200.145 €	-54.960 €
T1	122.736 €	153.632 €	-30.895 €
<b>2018</b>	<b>605.417 €</b>		
T4	122.144 €		
T3	129.497 €		
T2	200.145 €		
T1	153.632 €		
<b>Total</b>	<b>3.699.103 €</b>	<b>1.506.969 €</b>	<b>2.192.134 €</b>

Comp € SAMPERLY por Año y Trimestre

Trimestre ● T1 ● T2 ● T3 ● T4





# Acumulado Desde, Hasta o Entre fechas

---

Usaremos las funciones **CALCULATE + DATESBETWEEN**

Así como las funciones **BLANK, TODAY y SELECTEDVALUE.**





# Función DATESBETWEEN

**DATESBETWEEN**(<Dates>, <StartDate>, <EndDate>)

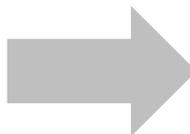
**DATESBETWEEN** (Fechas, Fecha Inicio, Fecha Fin)

*Columna de fechas = Calendario[Date]*

Devuelve una columna de fechas entre una fecha de inicio y una fecha de fin.

**EVALUATE**

```
DATESBETWEEN(  
    CALENDARIO[Date],  
    DATE(2019,1,28),  
    DATE(2019,2,3)  
)
```



Resultados

Resultado 1 de 1 ▾

	CALENDARIO[Date]
1	1/28/2019 12:00:00 AM
2	1/29/2019 12:00:00 AM
3	1/30/2019 12:00:00 AM
4	1/31/2019 12:00:00 AM
5	2/1/2019 12:00:00 AM
6	2/2/2019 12:00:00 AM
7	2/3/2019 12:00:00 AM



# Función DATESBETWEEN

---

## **DATESBETWEEN** (Fechas, Fecha Inicio, Fecha Fin)

Podemos sustituir la Fecha ***Inicio*** por la función **BLANK()**, función sin argumentos que devuelve un valor vacío.

En este caso, la función **DATESBETWEEN** devuelve una columna de fechas desde la primera fecha en nuestra tabla Calendario hasta la Fecha Fin que especifiquemos.

Asimismo, podemos sustituir la Fecha ***Fin*** por la función **BLANK()**

En este caso, la función **DATESBETWEEN** devuelve una columna de fechas desde la fecha especificada hasta la última fecha en nuestra tabla Calendario.



# Función DATESBETWEEN

---

## **DATESBETWEEN** (Fechas, Fecha Inicio, Fecha Fin)

Podemos sustituir la Fecha **Inicio** por la función **TODAY()**, función sin argumentos que devuelve la fecha de **hoy**.

En este caso, la función **DATESBETWEEN** devuelve una columna de fechas desde **hoy** hasta la Fecha Fin que especifiquemos, *siempre que la fecha hoy se encuentre dentro de nuestra tabla Calendario*.

Asimismo, podemos sustituir la Fecha **Fin** por la función **TODAY()**

En este caso, la función **DATESBETWEEN** devuelve una columna de fechas desde la fecha especificada hasta la fecha de **hoy**, *siempre que la fecha hoy se encuentre dentro de nuestra tabla Calendario*.



# Función SELECTEDVALUE

---

**SELECTEDVALUE** (<columnName>[, <alternateResult>])

**SELECTEDVALUE** (Columna [, Resultado alternativo])

Devuelve el valor de una columna seleccionado por el contexto de filtro. **Dicho contexto puede venir de un filtro propiamente o de un visual.**

Por ejemplo: el siguiente visual selecciona una fecha de inicio y una fecha de fin de la columna “Date” de nuestra tabla Calendario.



# Función SELECTEDVALUE



Uds Entre SelectedVal =

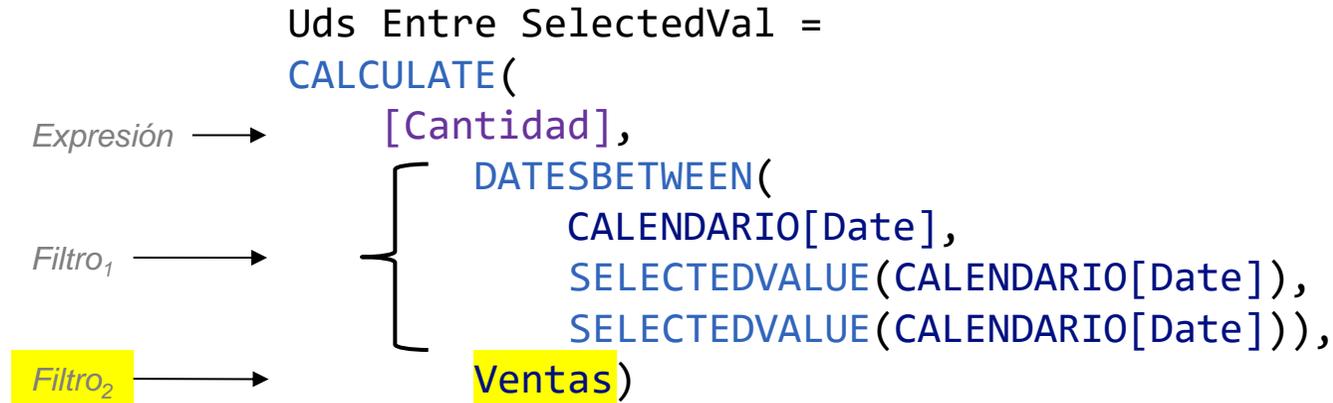
```

CALCULATE(
    [Cantidad],
    DATESBETWEEN(
        CALENDARIO[Date],
        // DATE(2018,8,22),DATE(2019,7,22) //Entre
        SELECTEDVALUE(CALENDARIO[Date]),SELECTEDVALUE(CALENDARIO[Date])),
    Ventas)
    
```

Liga segmentador con medida



# Función SELECTEDVALUE



**Nota:** En este caso, es necesario introducir un **segundo filtro** en la función **CALCULATE** especificando la tabla de hechos para la que se ha de realizar el cálculo.

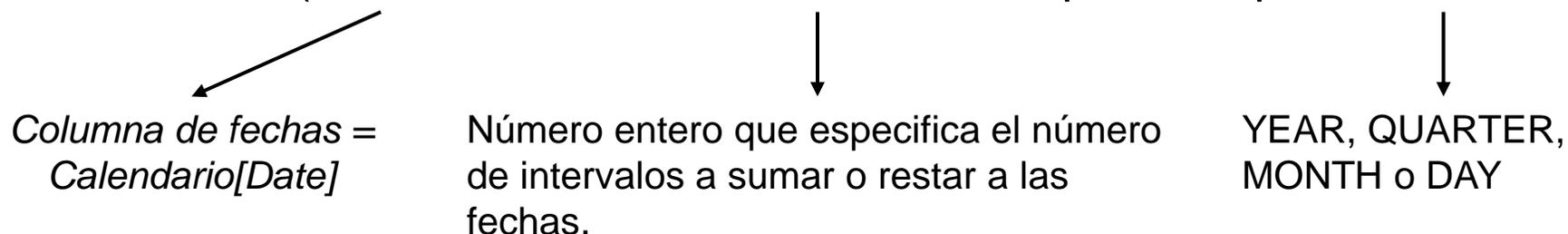


# Función DATEADD

---

**DATEADD** (<dates>,<number\_of\_intervals>,<interval>)

**DATEADD** (Fecha, Número de Intervalos a desplazar, Tipo de intervalo)



Devuelve una tabla que contiene una columna de fechas que se han desplazado hacia delante o hacia atrás en el tiempo según el número especificado de intervalos desde las fechas del contexto actual.



# Función DATEADD

**DATEADD** (Fecha, Número de Intervalos a desplazar, Tipo de intervalo)

EVALUATE

DATEADD(CALENDARIO[Date], 2, YEAR)

Desplaza las fechas de la tabla Calendario 2 años hacia adelante; es decir, genera una columna de fechas empezando con la primera fecha de la tabla Calendario + 2 años y termina con la última fecha de la tabla Calendario:



	CALENDARIO[Date]
1	1/1/2020 12:00:00 AM
2	1/2/2020 12:00:00 AM
3	1/3/2020 12:00:00 AM
4	1/4/2020 12:00:00 AM
5	1/5/2020 12:00:00 AM
6	1/6/2020 12:00:00 AM
7	1/7/2020 12:00:00 AM
8	1/8/2020 12:00:00 AM

359	12/24/2020 12:00:00 AM
360	12/25/2020 12:00:00 AM
361	12/26/2020 12:00:00 AM
362	12/27/2020 12:00:00 AM
363	12/28/2020 12:00:00 AM
364	12/29/2020 12:00:00 AM
365	12/30/2020 12:00:00 AM
366	12/31/2020 12:00:00 AM



# Función RELATED

---

**RELATED** (<column>)

**RELATED** (Columna)



Columna que contiene los valores que deseamos recuperar.

La función RELATED requiere que haya una relación entre la tabla actual y la tabla con información relacionada.

La función RELATED devuelve los valores de una tabla relacionada que se encuentre en el lado "uno" de la relación.

# Crear una tabla manualmente

Bicicletas Demvrek RESUELTO Año Fiscal • Guardado por última vez: Hoy a las 17:20

Archivo **Inicio** Ayuda Herramientas de tablas

Pegar Cortar Copia

Obtener Libro de datos Excel Centro de datos de OneLake SQL Server Introducir datos

Portapapeles Datos

Crear tabla

Columna1	+
1	
+	

Nombre:

Cargar Editar Cancelar



# Crear una tabla manualmente

Bicicletas Demvrek RESUELTO Año Fiscal • Guardado por última vez: Hoy a las 17:20

Archivo **Inicio** Ayuda Herramientas de tablas

Pegar Cortar Copia

Obtener Libro de datos Excel Centro de datos de OneLake SQL Server Introducir datos

Portapapeles Datos

Crear tabla

MesCalend	MesFiscal
1	5
2	6
3	7
4	8
5	9
6	10
7	11
8	12
9	1
10	2
11	3
12	4

Nombre:

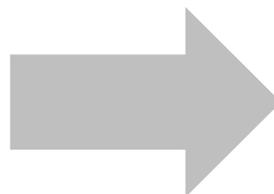
Cargar Editar Cancelar



# Función RELATED

MesFisc = RELATED(Correspondencia[MesFiscal])

Date	Año	Mes	MesFisc
01/01/2018 0:00:00	2018	ene	5
02/01/2018 0:00:00	2018	ene	5
03/01/2018 0:00:00	2018	ene	5
04/01/2018 0:00:00	2018	ene	5
05/01/2018 0:00:00	2018	ene	5
06/01/2018 0:00:00	2018	ene	5
07/01/2018 0:00:00	2018	ene	5
08/01/2018 0:00:00	2018	ene	5
09/01/2018 0:00:00	2018	ene	5
10/01/2018 0:00:00	2018	ene	5
11/01/2018 0:00:00	2018	ene	5
12/01/2018 0:00:00	2018	ene	5
13/01/2018 0:00:00	2018	ene	5
14/01/2018 0:00:00	2018	ene	5
15/01/2018 0:00:00	2018	ene	5
16/01/2018 0:00:00	2018	ene	5
17/01/2018 0:00:00	2018	ene	5
18/01/2018 0:00:00	2018	ene	5



MesCalend	MesFiscal
1	5
2	6
3	7
4	8
5	9
6	10
7	11
8	12
9	1
10	2
11	3
12	4



# Función SWITCH

---

- Devuelve los resultados determinados en el/los argumento(s) “Resultado” según coincida el valor de cada elemento de una columna con el argumento “Valor” de la función SWITCH.
- Si el valor de del elemento en la columna no coincide con ningún argumento “Valor” dado a la función, entonces devuelve el resultado determinado en el argumento “de lo contrario”.
- Si dicho argumento no se especifica, entonces, devuelve un espacio en blanco equivalente al resultado de la función BLANK.
- ***Es equivalente a una función SI anidada en sí misma varias veces.***

Es útil cuando queremos obtener un porcentaje por categoría. Por ejemplo:

- Para cada subcategoría se ha de calcular el % sobre su categoría;
- Para cada categoría se ha de calcular el porcentaje sobre las ventas totales.

Este es el concepto de “porcentaje sobre [una categoría] *padre*”.



# Función SWITCH

**SWITCH** (<expression>, <value>, <result>[, <value>, <result>]...[, <else>])

**SWITCH** (Expresión, Valor, Resultado [, Valor, Resultado]...[, de lo contrario])

←  
Cualquier expresión  
DAX que devuelva un  
**único valor escalar**.

Par “valor-resultado”:

- **Valor:** Valor del elemento de la columna de datos a los que se aplica la función SWITCH que se va a comparar con el resultado de “Expresión”.
- **Resultado:** El resultado de la función switch para cada elemento de la columna de datos a los que se aplica la función SWITCH si:

*Valor = resultado de “Expresión”.*

- Este par se puede repetir varias veces dentro de la función SWITCH.

↓  
El resultado de la  
función switch para cada  
elemento de la columna  
de datos a los que se  
aplica la función  
SWITCH si:

*Valor <> resultado de  
“Expresión”.*



# Función SWITCH

EVALUATE

ADDCOLUMNS(

CALENDAR(DATE(YEAR(MIN(Ventas[Fecha])),1,1),

DATE(YEAR(MAX(Ventas[Fecha])),12,31)

),

"Año", YEAR([Date]),

// "Trimestre", "T"&QUARTER([Date]),

"No. Mes", MONTH([Date]),

"Mes", FORMAT([Date],"mmm"),

"TrimFiscal", "TF"&

SWITCH(MONTH([Date]),

9,1, 10,1, 11,1,

12,2, 1,2, 2,2,

3,3, 4,3, 5,3,

6,4, 7,4,

4

)

)

← Expresión DAX que devuelve un **único valor escalar** para cada fila (el número del mes).

← Pares "Valor-resultado"

← El resultado de la función switch si *Valor* <> resultado de "Expresión".



# Función SWITCH

## DEFINE MEASURE

```
MEDIDAS[ImporteDescontado]=
```

```
SUMX(VALUE(Productos[ Categoría ]),
```

```
VAR PctDto=  
SWITCH(Productos[ Categoría ],
```

```
"Bicicletas", 0.05,  
"Componentes", 0.10,  
"Accesorios", 0.12,  
"Ropa", 0.15,  
0)
```

Equivalencia a una función SI anidada en sí misma varias veces.

```
RETURN [Importe]*(1-PctDto)
```

```
)
```

Definimos la medida “ImporteDescontado” y la incluimos en la tabla “MEDIDAS”.

El argumento “tabla” de SUMX es una lista de las categorías de producto sin repetir (VALUES)

SWITCH toma el valor de cada elemento de la columna “Categoría” de la tabla “Productos” y:

- Si dicho valor = “Bicicletas”, le asigna un 5%
- Si dicho valor = “Componentes”, le asigna 10%
- Si dicho valor = “Accesorios”, le asigna un 12%
- Si dicho valor = “Ropa”, le asigna un 15%
- A cualquier otro valor le asigna un 0% de dto.

[ ImporteDescontado ] nos devuelve el importe de ventas menos los descuentos agrupado (SUMX) **por categoría**.

	Productos[Categoría]	[Importe]	[Importe Descontado]
1	Bicicletas	3601049.17	3420996.71
2	Accesorios	66063.83	58136.17
3	Ropa	31989.7	27191.24



# Función ISINSCOPE

---

Una forma intuitiva de ver el funcionamiento de esta función es la siguiente:

La función ISINSCOPE:

- Evalúa cada **fila** de una tabla **resultante**.
- Devuelve el valor booleano VERDADERO si en dicha fila encuentra un valor que esté dentro del ámbito de la columna especificada como argumento de la función.

ISINSCOPE(<columnName>)

**ISINSCOPE** (Columna)



Columna que forma parte de una jerarquía (explícita o implícita)



# Función ISINSCOPE

Filas	
Año Jerarquía	∨ ×
Año	×
Trimestre	×
MesNombre	×
Día	×

Jerarquía explícita

¿Hay algún elemento en esta fila que esté dentro del ámbito ---- de la tabla Calendario?

NivelMes = ISINSCOPE(CALENDARIO[Mes])

NivelTrim = ISINSCOPE(CALENDARIO[Trimestre])

NivelAño = ISINSCOPE(CALENDARIO[Año])

Año	Cantidad X	Mes	Trim.	Año
☐ 2018	25.140 uds	No	No	Sí
☐ T1	4.855 uds	No	Sí	Sí
☐ ene	1.664 uds	Sí	Sí	Sí
☐ feb	1.506 uds	Sí	Sí	Sí
☐ mar	1.685 uds	Sí	Sí	Sí
☐ T2	6.428 uds	No	No	Sí
☐ T3	6.907 uds			
☐ T4	6.950 uds			
☐ 2019	224.841 uds		...	
☐ 2020	360.679 uds			
Total	610.660 uds	No	No	No



Año	Cantidad X	NivelMes	NivelTrim	NivelAño
☐ 2018	25.140 uds	False	False	True
☐ T1	4.855 uds	False	True	True
☐ ene	1.664 uds	True	True	True
☐ feb	1.506 uds	True	True	True
☐ mar	1.685 uds	True	True	True
☐ T2	6.428 uds	False	True	True
☐ T3	6.907 uds	False	True	True
☐ T4	6.950 uds	False	True	True
☐ 2019	224.841 uds	False	False	True
☐ 2020	360.679 uds	False	False	True
☐ 2021		False	False	True
Total	610.660 uds	False	False	False





# Contenidos

---

1. Modelo de datos
2. Transformación de datos
3. DAX
4. Modelo de datos - Ambigüedad

---

# Modelo

## Ambigüedad en el modelo



## Doble filtro

Matriz que resume el importe de ventas por colores y año.

Color	2007	2008	2009	Total
Amarillo	1.314 €	4.028 €	284 €	<b>5.625 €</b>
Azul	120.713 €	67.184 €	25.452 €	<b>213.348 €</b>
Blanco	192.198 €	251.743 €	232.690 €	<b>676.631 €</b>
Dorado	7.827 €	15.707 €	1.740 €	<b>25.274 €</b>
Gris	180.994 €	192.143 €	67.871 €	<b>441.008 €</b>
Gris plateado	33.161 €	10.763 €	7.476 €	<b>51.400 €</b>
Lila	190 €		61 €	<b>251 €</b>
Marrón	60.429 €	52.188 €	65.349 €	<b>177.967 €</b>
Naranja	12.624 €	17.688 €	15.524 €	<b>45.835 €</b>
Negro	257.528 €	221.772 €	118.773 €	<b>598.073 €</b>
Plateado	396.711 €	194.851 €	116.082 €	<b>707.645 €</b>
Rojo	51.947 €	24.343 €	25.164 €	<b>101.454 €</b>
Rosa	64.196 €	17.787 €	10.525 €	<b>92.508 €</b>
Transparente	109 €	72 €	107 €	<b>288 €</b>
Turquesa	3.890 €	4.062 €	2.300 €	<b>10.253 €</b>
Verde	75.384 €	48.204 €	16.516 €	<b>140.104 €</b>
<b>Total</b>	<b>1.459.216 €</b>	<b>1.122.535 €</b>	<b>705.914 €</b>	<b>3.287.665 €</b>

En el caso de este visual, actúan **2 filtros** sobre la tabla de ventas:

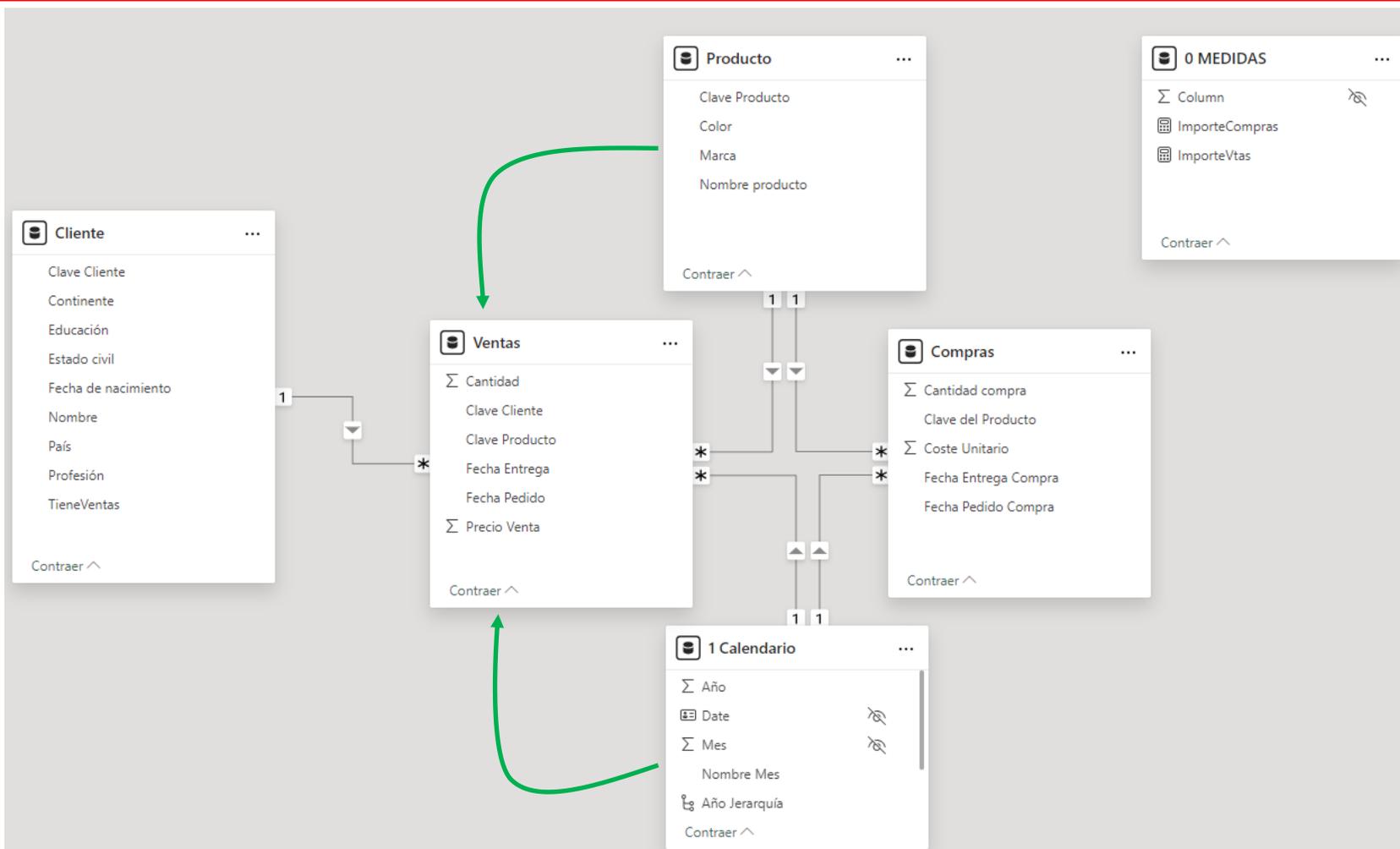
- 1.CALENDARIO - Año
- 2.PRODUCTO - Color

Las tablas de dimensiones “CALENDARIO” y “PRODUCTO”, ambas se relacionan directamente con la tabla “Ventas”.

Asimismo, los filtros de las relaciones entre estas tablas y la tabla “Ventas” van en dirección hacia la tabla ventas; luego, ambos filtros **actúan** sobre la tabla ventas y el **doble filtro** funciona.



# Doble filtro





# Filtro en dirección prohibida

Matriz que resume cuántas marcas hemos vendido en cada país.

País	Recuento de Marca
Alemania	11
Armenia	11
Australia	11
Bután	11
Canadá	11
China	11
España	11
Estados Unidos	11
Francia	11
Grecia	11
India	11
Irán	11
Irlanda	11
Italia	11
Japón	11
Kirguistán	11
Korea del Sur	11
<b>Total</b>	<b>11</b>

En este visual intentamos ir de la tabla Cliente a la tabla Producto PERO...

Las direcciones de filtro van:

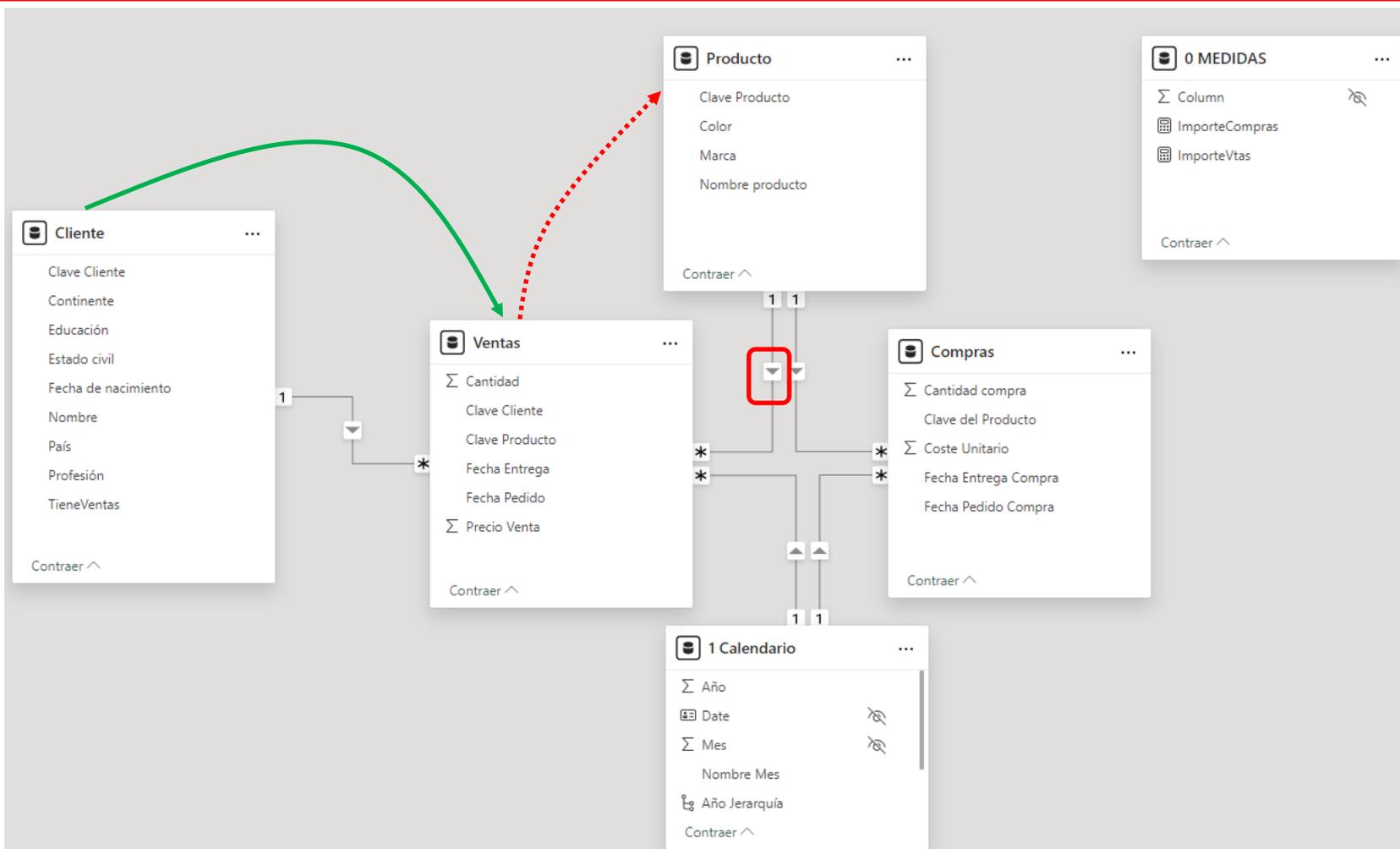
1. De Cliente a Ventas
2. De Producto a Ventas (NO de VENTAS a PRODUCTO)

Siendo así, el filtro **no puede viajar** de VENTAS a PRODUCTO puesto que la dirección del filtro para estas dos tablas sólo va en una dirección: DE Producto a Ventas.

Luego, para este visual el filtro **NO** funciona.



# Doble filtro





# Filtro en dirección prohibida

Lo mismo nos ocurre con los segmentadores y con la tabla de compras por cliente:

Nombre  
Adams, Stephanie

- Color
- Amarillo
  - Azul
  - Blanco
  - Dorado
  - Gris
  - Gris plateado
  - Lila
  - Marrón
  - Naranja
  - Negro
  - Plateado
  - Rojo
  - Rosa
  - Transparente
  - Turquesa
  - Verde

Año	ImporteVtas
2007	1.013 €
ene	1.013 €
<b>Total</b>	<b>1.013 €</b>

Nombre  
Adams, Sean

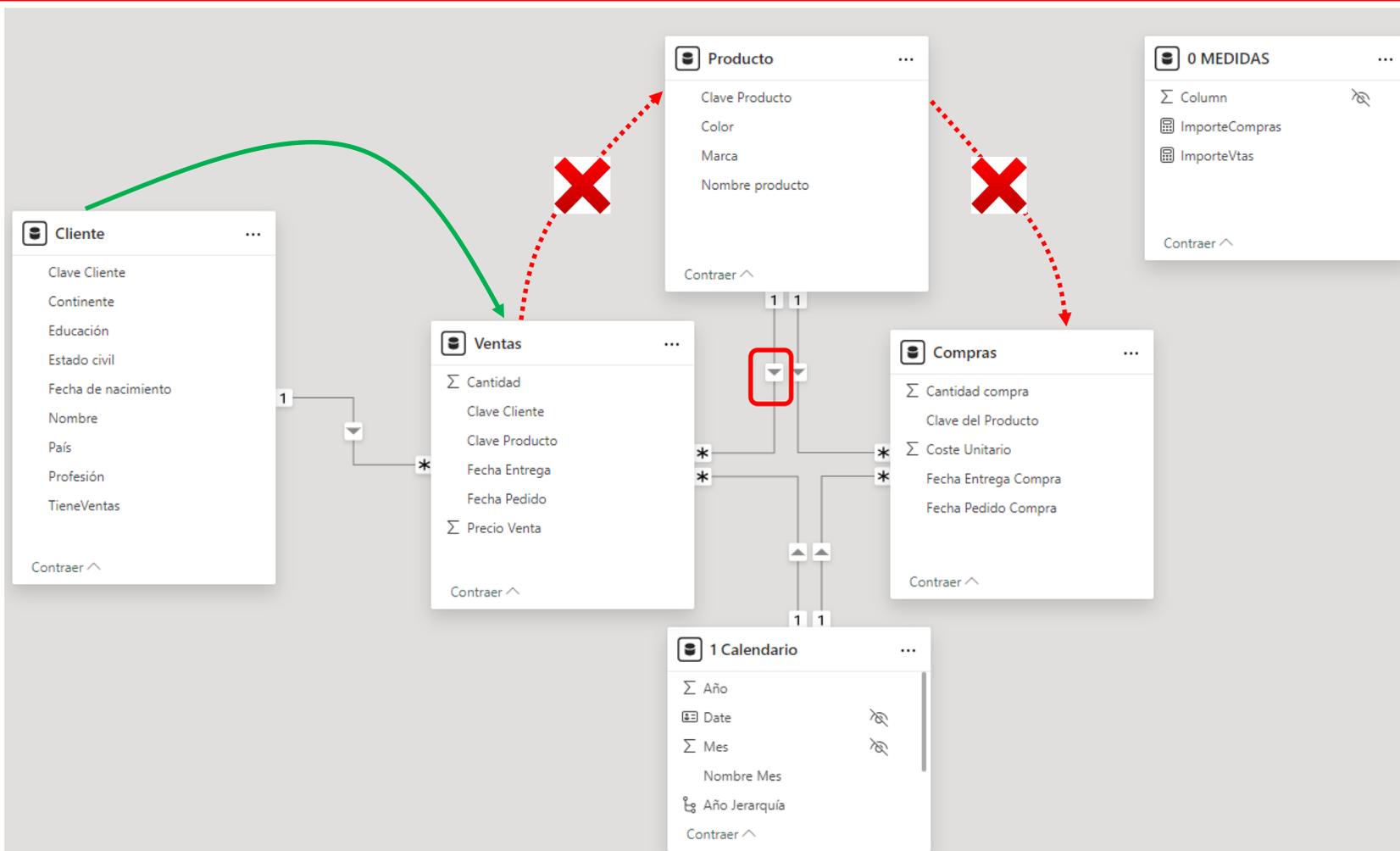
- Color
- Amarillo
  - Azul
  - Blanco
  - Dorado
  - Gris
  - Gris plateado
  - Lila
  - Marrón
  - Naranja
  - Negro
  - Plateado
  - Rojo
  - Rosa
  - Transparente
  - Turquesa
  - Verde

Año	ImporteVtas
2007	4.363 €
nov	4.363 €
<b>Total</b>	<b>4.363 €</b>

Nombre	ImporteCompras
Adams, Aaron	2.823.617 €
Adams, Adam	2.823.617 €
Adams, Alex	2.823.617 €
Adams, Alexandra	2.823.617 €
Adams, Allison	2.823.617 €
Adams, Amanda	2.823.617 €
Adams, Amber	2.823.617 €
Adams, Andrea	2.823.617 €
Adams, Angel	2.823.617 €
Adams, Bailey	2.823.617 €
Adams, Ben	2.823.617 €
<b>Total</b>	<b>2.823.617 €</b>

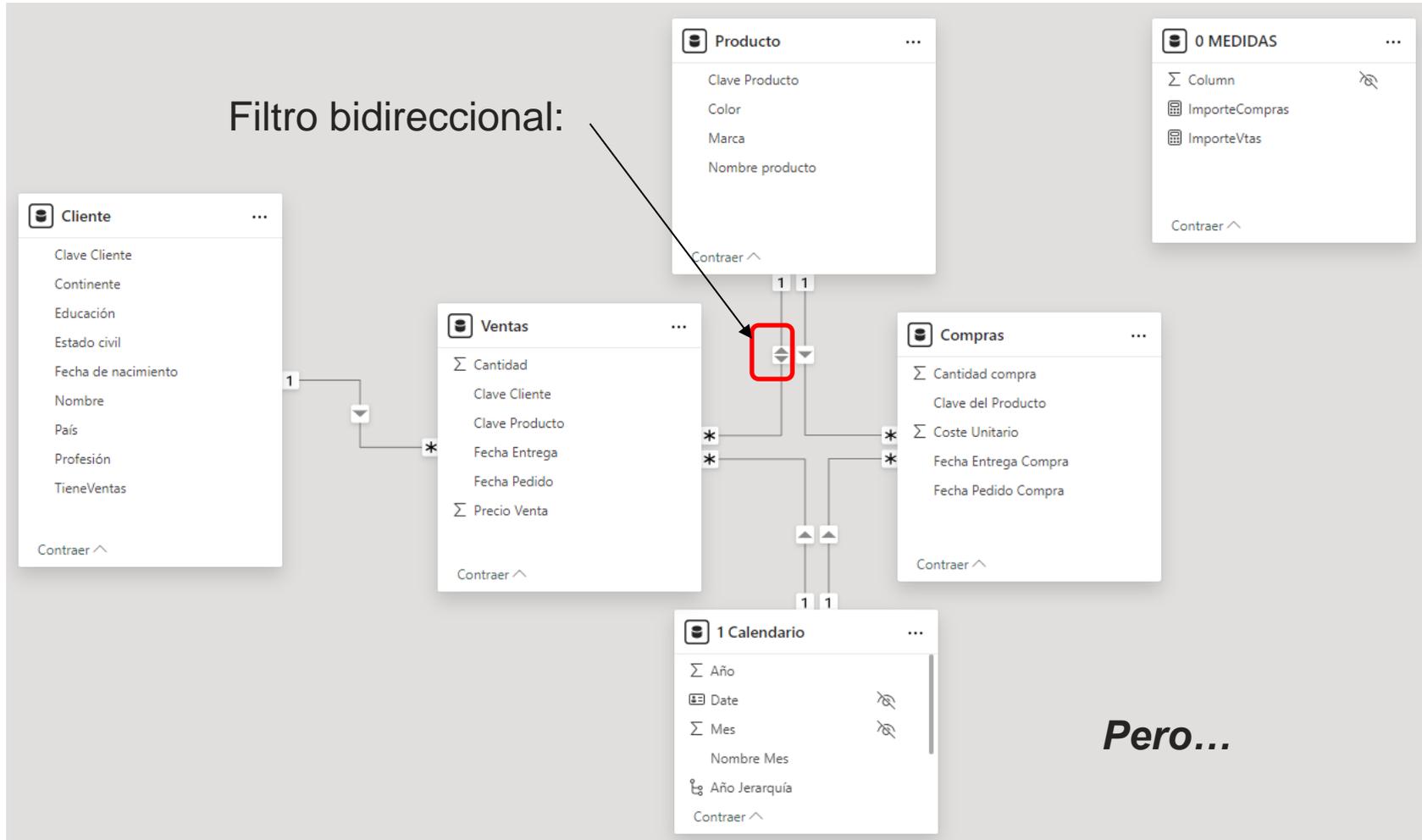
**Los filtros no funcionan.**

# Filtro en dirección prohibida

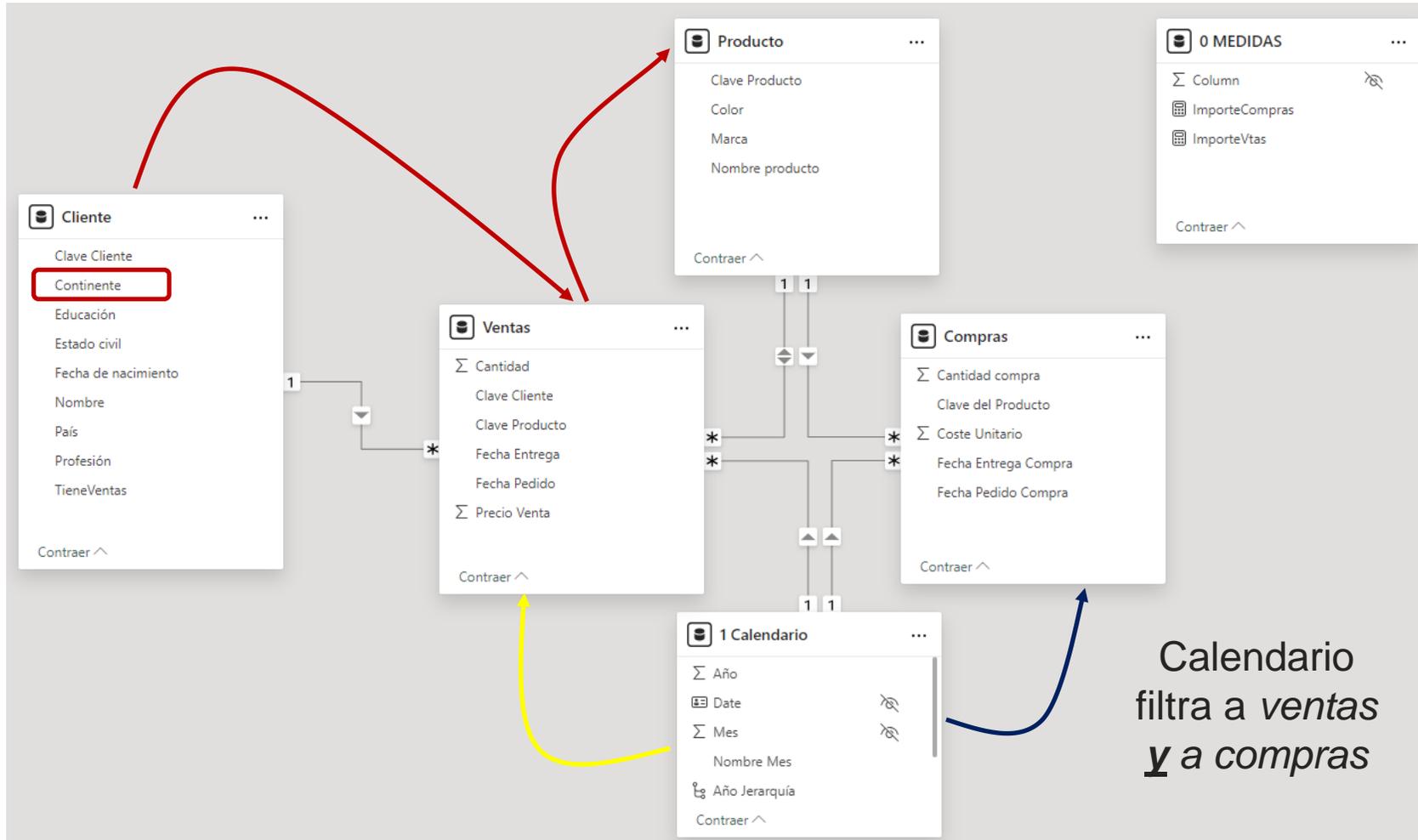




# Solución 1



# Ambigüedad



Calendario  
filtra a *ventas*  
y a *compras*



# Ambigüedad

---

Doble filtro sobre tabla ventas:

- Cliente[Continente] filtra a Ventas
- Calendario[Año] filtra a Ventas

Ventas filtra a Producto  
Producto filtra a Compra

y además...

## Calendario filtra a **Ventas** y a Compras

Continente	2007	2008	2009	<b>Total</b>
Asia	826.345 €	386.677 €	245.270 €	<b>1.612.549 €</b>
Europa	709.301 €	218.653 €	243.296 €	<b>1.297.167 €</b>
Norteamérica	682.498 €	469.291 €	187.941 €	<b>1.436.890 €</b>
<b>Total</b>	<b>1.240.334 €</b>	<b>983.256 €</b>	<b>800.026 €</b>	<b>3.023.617 €</b>

# Ambigüedad



Calendario sólo filtra a *ventas*

Eliminamos esta relación mediante la función **CROSSFILTER**



# Ambigüedad

---

Doble filtro sobre tabla ventas:

- Cliente[Continente] filtra a Ventas
- Calendario[Año] filtra a Ventas

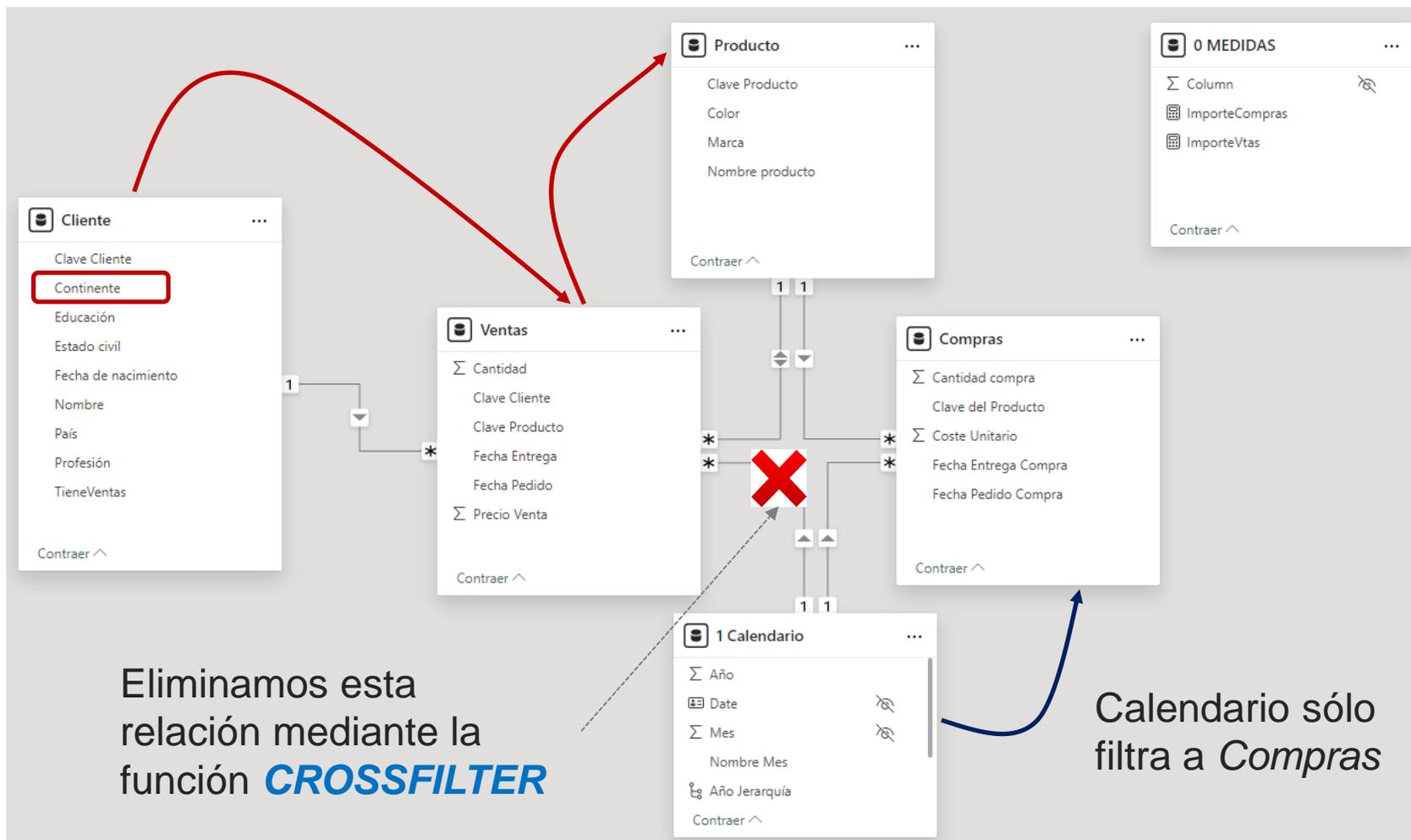
Ventas filtra a Producto  
Producto filtra a Compra

y además...

## Calendario filtra sólo a **Ventas**

Continente	2007	2008	2009	Total
Asia	938.959,92	490.181,22	332.381,58	<b>1.612.548,63</b>
Europa	804.667,66	297.650,35	323.114,43	<b>1.297.167,22</b>
Norteamérica	771.980,27	563.854,39	236.432,54	<b>1.436.889,82</b>
<b>Total</b>	<b>1.497.023,00</b>	<b>1.244.756,67</b>	<b>800.870,95</b>	<b>3.023.616,54</b>

# Ambigüedad





# Ambigüedad

---

Doble filtro sobre tabla ventas:

- Cliente[Continente] filtra a Ventas
- Calendario[Año] filtra a Ventas

Ventas filtra a Producto  
Producto filtra a Compra

y además...

## Calendario filtra sólo a **Compras**

Continentes	2007	2008	2009	Total
Asia	875.994,21	441.740,79	294.813,63	<b>1.612.548,63</b>
Europa	739.447,10	273.054,99	284.665,12	<b>1.297.167,22</b>
Norteamérica	716.710,75	494.950,11	225.228,96	<b>1.436.889,82</b>
<b>Total</b>	<b>1.240.333,55</b>	<b>983.256,49</b>	<b>800.026,49</b>	<b>3.023.616,54</b>

# Ambigüedad

Doble filtro sobre tabla ventas:

- Cliente[Continente] filtra a Ventas
- Calendario[Año] filtra a Ventas

Ventas filtra a Producto  
Producto filtra a Compra

y además...



¡Los tres resultados son distintos!

## Calendario filtra a Ventas y a Compras

Continente	2007	2008	2009	Total
Asia	826.345 €	386.677 €	245.270 €	<b>1.612.549 €</b>
Europa	709.301 €	218.653 €	243.296 €	<b>1.297.167 €</b>
Norteamérica	682.498 €	469.291 €	187.941 €	<b>1.436.890 €</b>
<b>Total</b>	<b>1.240.334 €</b>	<b>983.256 €</b>	<b>800.026 €</b>	<b>3.023.617 €</b>

## Calendario filtra sólo a Ventas

Continente	2007	2008	2009	Total
Asia	938.959,92	490.181,22	332.381,58	<b>1.612.548,63</b>
Europa	804.667,66	297.650,35	323.114,43	<b>1.297.167,22</b>
Norteamérica	771.980,27	563.854,39	236.432,54	<b>1.436.889,82</b>
<b>Total</b>	<b>1.497.023,00</b>	<b>1.244.756,67</b>	<b>800.870,95</b>	<b>3.023.616,54</b>

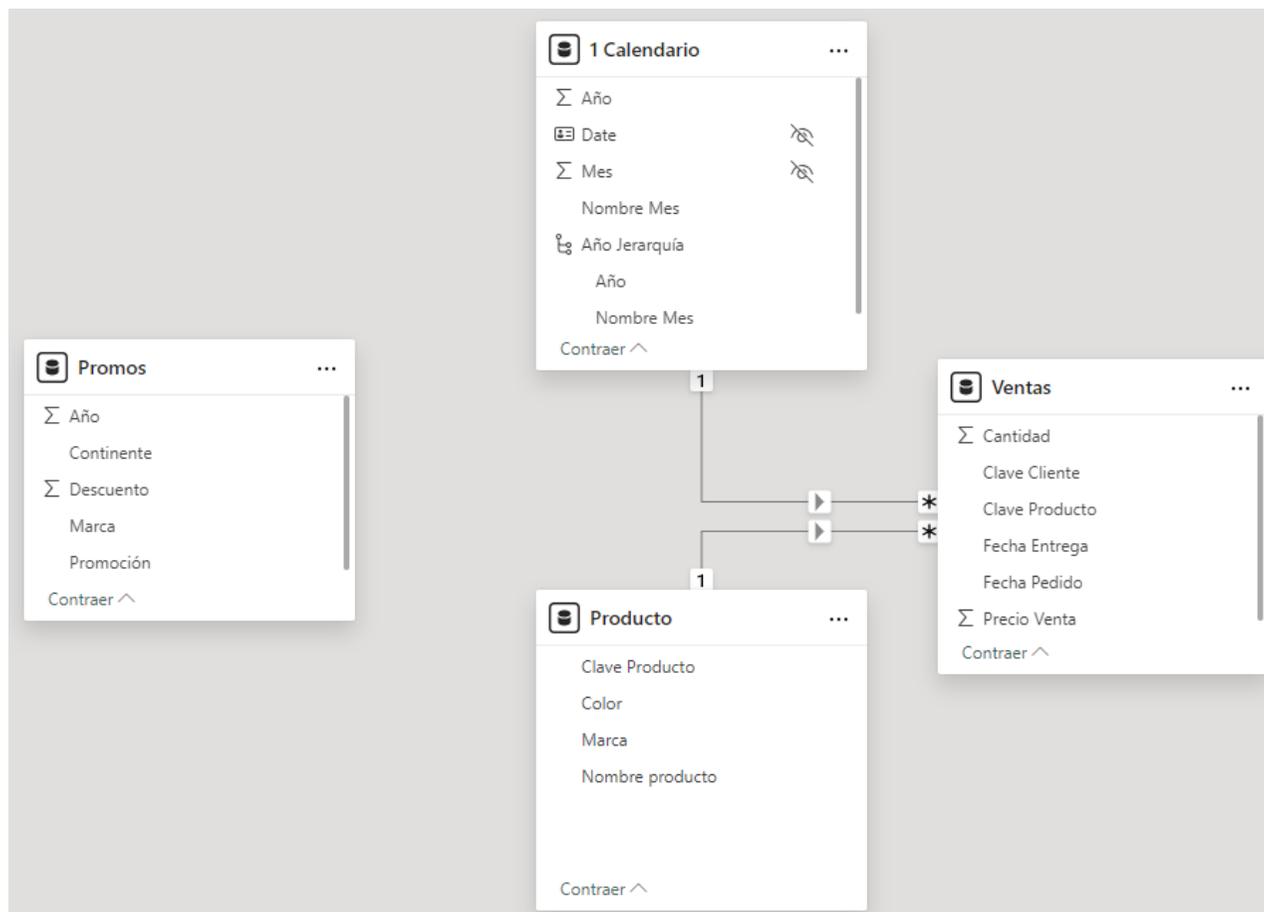
## Calendario filtra sólo a Compras

Continente	2007	2008	2009	Total
Asia	875.994,21	441.740,79	294.813,63	<b>1.612.548,63</b>
Europa	739.447,10	273.054,99	284.665,12	<b>1.297.167,22</b>
Norteamérica	716.710,75	494.950,11	225.228,96	<b>1.436.889,82</b>
<b>Total</b>	<b>1.240.333,55</b>	<b>983.256,49</b>	<b>800.026,49</b>	<b>3.023.616,54</b>

# Relación virtual

# Relación virtual

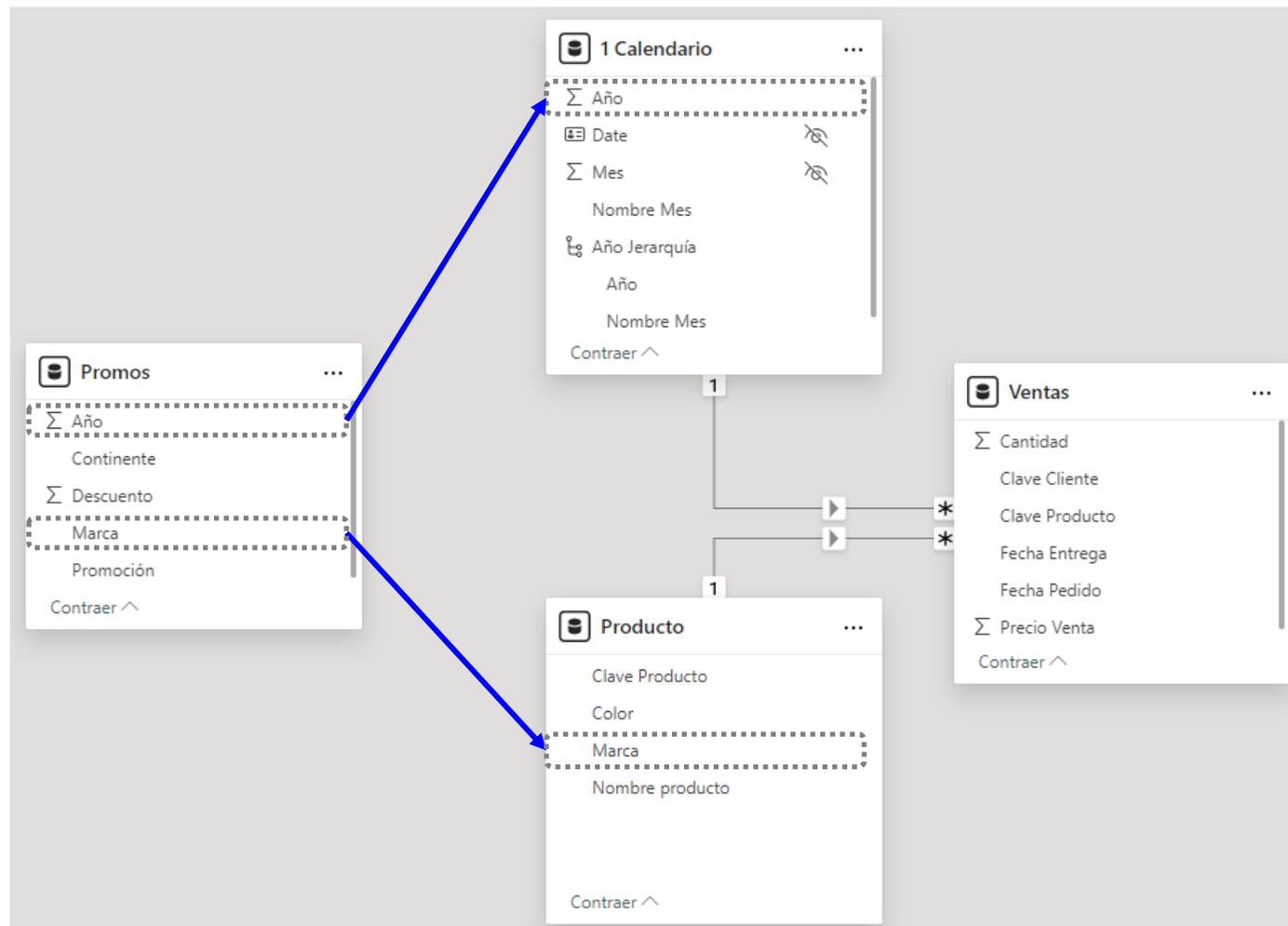
Visualizar importe de ventas CON y SIN descuento por marca y año.



# Relación virtual

Lo anterior sería sencillo de hacer estableciendo las siguientes relaciones.

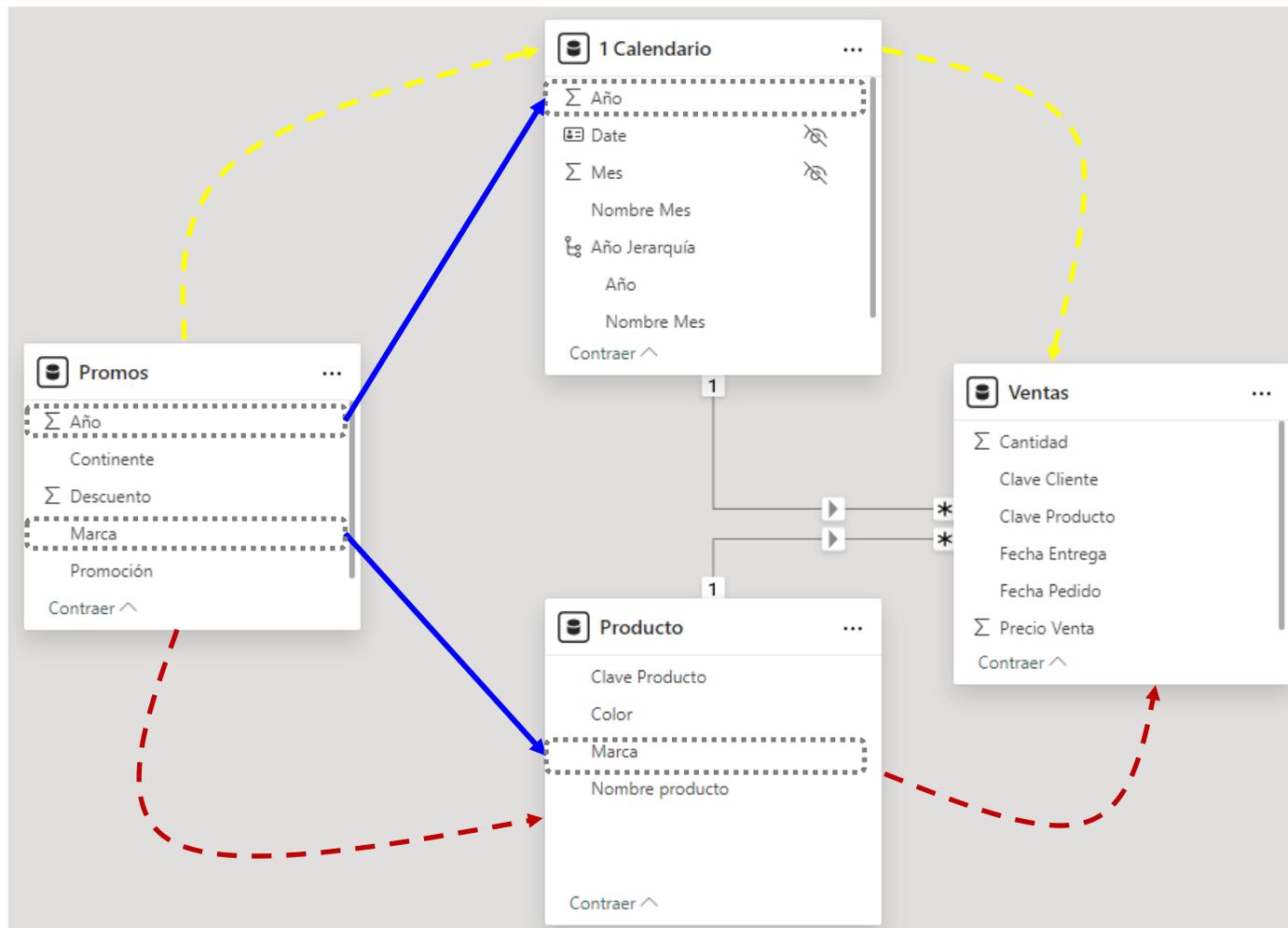
El problema es que, si hacemos eso, generaríamos ambigüedad en el modelo...



# Relación virtual – Problema

Lo anterior sería sencillo de hacer estableciendo las siguientes relaciones.

El problema es que, si hacemos eso, generaríamos ambigüedad en el modelo.





# Relación virtual – Solución

---

Crear una relación virtual mediante la función **TREATAS**.

**TREATAS** (table\_expression, <column>[, <column>[, <column>[,...]]] )

**TREATAS** (Expresión de tabla, Columna existente [, Columna existente [,...]] )

La función TREATAS devuelve una tabla que contiene todas las filas de las columnas **que también están** en la expresión de tabla.

Trata a las columnas de la tabla creada mediante la “Expresión de tabla” **como si fueran** columnas de otras tablas del modelo semántico (es decir, del modelo de datos).

Es decir, asigna a las columnas de la tabla creada mediante “Expresión de tabla”, la misma “*categoría*”, el mismo linaje (*data lineage*) que las columnas de otras tablas del modelo semántico.

# Función TREATAS

---

**TREATAS** (Expresión de tabla, Columna existente [, Columna existente [,...]] )

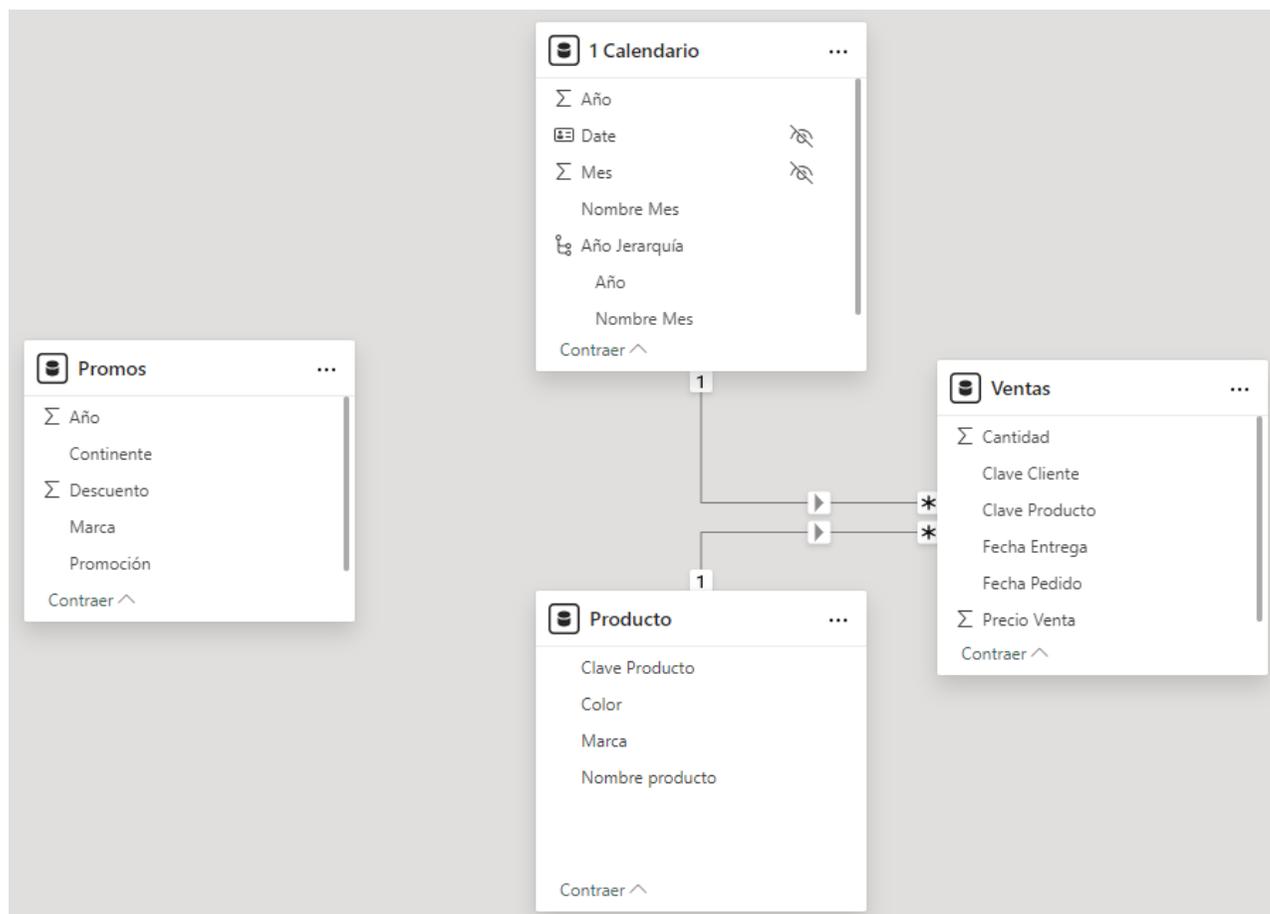
Expresión DAX que devuelva una **tabla**.

Columnas(s) existente(s) en otra(s) tabla(s) de datos del modelo semántico.

El número de columnas especificado debe coincidir con el número de columnas de la expresión de tabla y estar en el mismo orden.

# Relación virtual

¿Qué queremos hacer?



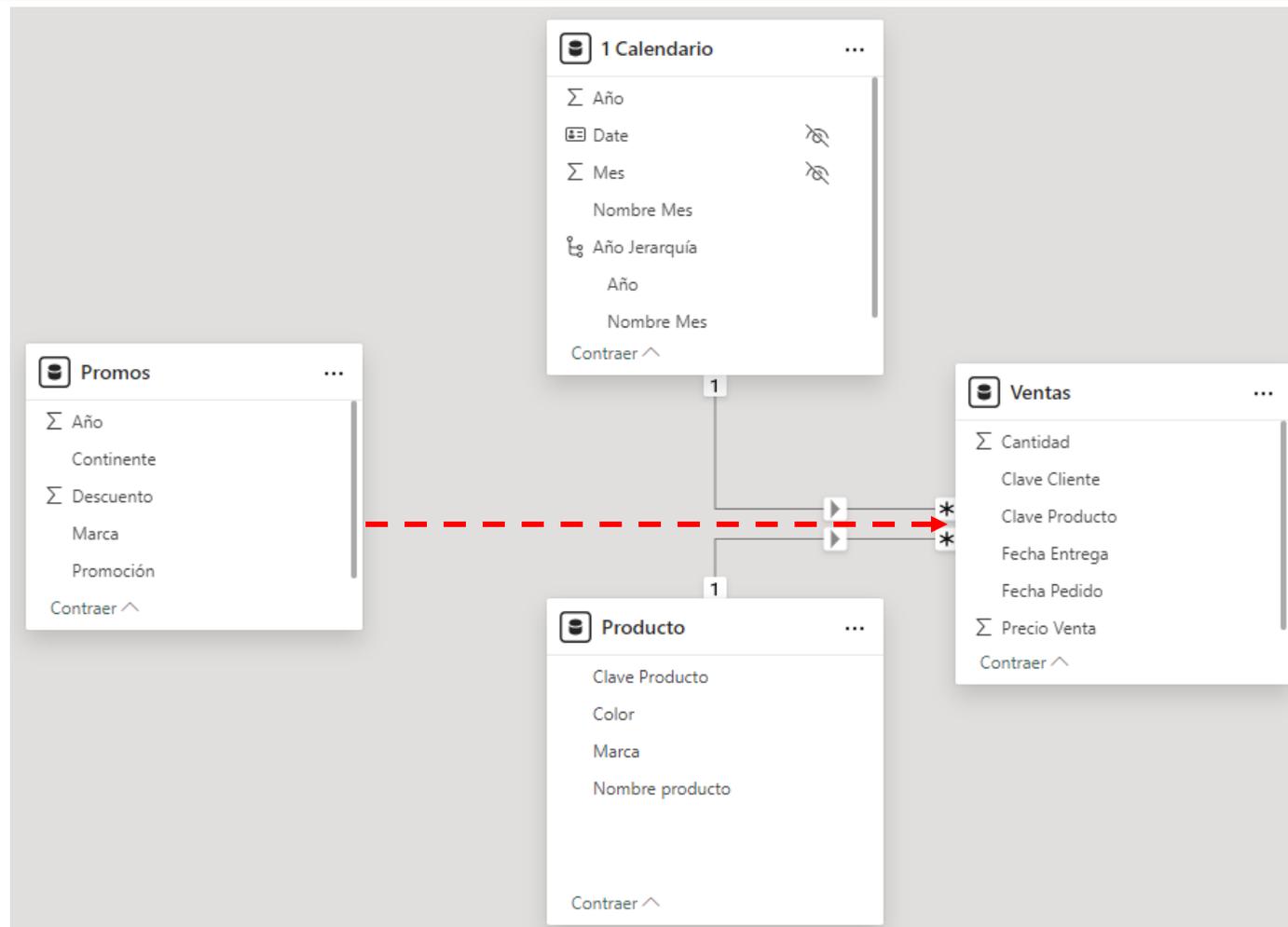
Queremos crear una tabla (intermedia) que, a su vez, sea capaz de filtrar la tabla “Ventas” mediante el uso de DAX.

Es decir...

# Relación virtual – Solución no viable

Podríamos crear un filtro que viajase de Promos a Ventas.

No obstante, para ello y aunque no vayamos relacionar las tablas en el modelo de datos, seguimos necesitando una columna común entre las dos tablas para poder relacionarlas.

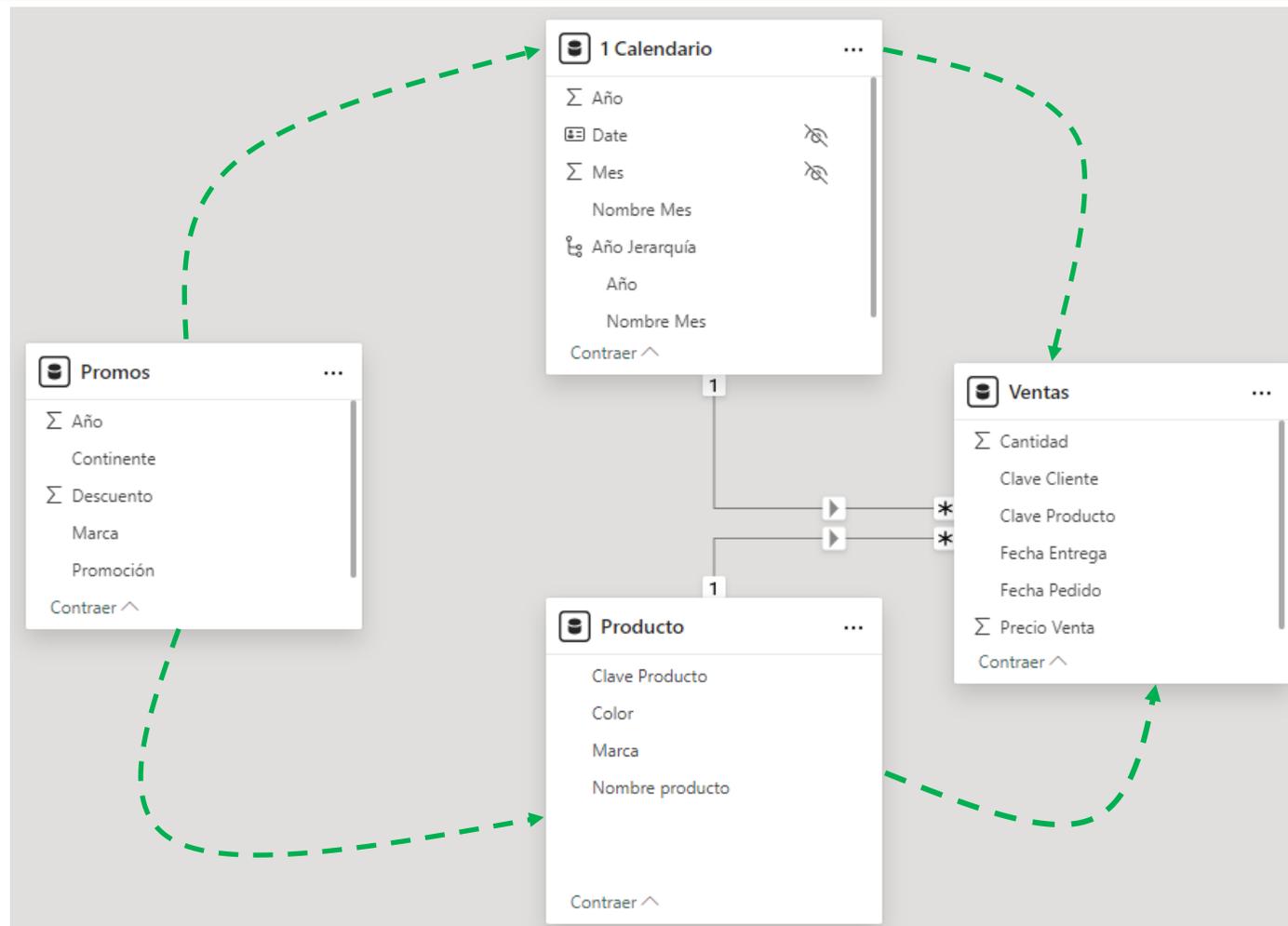


# Relación virtual – Solución

Queremos crear *sólo para esta ocasión*, un **dobles filtro** sobre la tabla de ventas que viaje:

- De Promos a Calendario a Ventas
- y
- De Promos a Producto a Ventas

**¿Cómo?...**





# Relación virtual – Solución

DEFINE

```
MEASURE '0 MEDIDAS' [VtasCONDesc] =
```

```
  CALCULATE (
```

```
    [ImporteVtas],
```

```
    SUMMARIZE (
```

```
      Promos,
```

```
      Promos[Marca],
```

```
      Promos[Año]
```

```
    )
```

```
  )
```

EVALUATE

```
SUMMARIZECOLUMNS (
```

```
  Promos[Promoción],
```

```
  Promos,
```

```
  "VentasCONPromo", [VtasCONDesc]
```

```
)
```

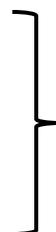


Tabla intermedia creada con objeto de filtrar los datos de la tabla "Ventas".

Hasta aquí, sólo hemos creado la tabla por la que queremos filtrar, pero NO hemos trasladado el filtro de esta tabla a la tabla Ventas:

	Promos[Promoción]	[VentasCONPromo]
1	Primavera	3287664.52
2	Navidad	3287664.52
3	Vuelta al cole	3287664.52
4	Verano	3287664.52

# Función TREATAS

