



Caso 141 Superstore UK

01 ETL y Modelo de Datos. Priorizar el uso del editor de consultas. Parámetro de conexión avanzado a carpeta y al fichero. Tabla calendario con lenguaje M. Consultas de referencia. Organizar consultas y parametros.

Jose Ignacio González Gómez

Departamento de Economía, Contabilidad y Finanzas - Universidad de La Laguna

www.jggomez.eu

V. 2.5

Ejercicio adaptado de: [Find Repeating Customers Using Cohort Analysis](#)

Contenido

Resumen y objetivos	2
Priorizar el uso del editor de consultas	2
Presentacion – Información disponible	3
PBI Basic Calendar Table	3
Superstore – Customer	3
Superstore – Sales	3
Superstore – Product	4
Se pide:	5
FASE I: ETL, Trabajando con Power Query	5
Priorizando el trabajo con el editor de consultas	5
Conexión al Dataframe	5
Organización de las consultas y parametros	10
Configurar tabla calendario en el editor de consultas con lenguaje M	10
Creando Consultas Referencia y ajustes	12
Diseño del modelo de datos v1 – Modelados y relaciones	16
Crear tabla Medidas	19
Modelo y relaciones	19
FASE II: Informe I PDTE DE DESARROLLO	19
Orientación y notas	20
Anexo. Bibliografía y conceptos técnicos relacionados	20
Crear tabla calendario con DAX o con lenguaje M	20
Crear la tabla calendario con DAX	20
Crear la tabla calendario con M (Power Query)	20
Tabla comparativa Calendario en DAX o en M	20
¿Cuál elegir?	21

Ejemplo usando la función DAX CalendarAuto ()	21
Crear columnas de fechas Fecha de Inicio y Fin de la Semana	22
Consultas de referencia.....	22
Concepto	22
Usos y ventajas.....	22
Ejemplos	23

Resumen y objetivos

FASE I

Justificar, priorizar el uso del editor de consultas

Trabajando con el editor de consultas (Power Query)

- Crear parámetro de conexión avanzado, a la carpeta contenedora y al fichero.
- Creando una tabla calendario en el editor de consultas con lenguaje M
- Trabajando con consultas de referencias
- Organizar consultas y parametros, creacion de grupos
- Creando agrupaciones en modo avanzado en consultas y creando columnas

Modelo semántico. Crear tabla calendario. Crear tablas para organizar las medidas.

Funciones DAX aplicadas: CALENDARAUTO,

- ComienzoSemana: =Calendario2[Date]-WEEKDAY(Calendario2[Date];2)+1
- FinDeLaSemana:= Calendario2[Date]-WEEKDAY(Calendario2[Date];2)+7

Priorizar el uso del editor de consultas

Técnicamente es que es mejor hacer todas las transformaciones posibles en el editor de consultas (PQ) que con fórmulas DAX en el modelo de datos por varias razones clave:

Separación de responsabilidades

- **PQ** se encarga de la **preparación de datos**: limpieza, filtrado, transformación, etc.
- **DAX** se usa para **modelado y análisis**: cálculos, medidas, KPIs, etc.

Mantener esta separación hace que el modelo sea más claro y fácil de mantener

Rendimiento

- PQ transforma los datos **antes de cargarlos al modelo**, lo que reduce el tamaño del modelo y mejora el rendimiento.
- DAX trabaja **sobre los datos ya cargados**, lo que puede ser más costoso en tiempo de cálculo, especialmente con grandes volúmenes de datos.

Reutilización de datos

- Las transformaciones en PQ pueden ser reutilizadas en múltiples tablas o informes.
- Las fórmulas DAX suelen ser específicas para un visual o medida.

Limpieza y calidad de datos

- PQ tiene herramientas más robustas para detectar y corregir errores, tipos de datos inconsistentes, valores nulos, etc.
- DAX no está diseñado para limpiar datos, sino para analizarlos.

Facilidad de uso

- PQ tiene una interfaz más intuitiva, con pasos visuales y editor de fórmulas M.
- DAX requiere más conocimiento técnico y puede ser más difícil de depurar.

Presentacion – Información disponible

Disponemos de la información básica del ERP de un hipermercado del Reino Unido y antes del diseño del cuadro de mando estamos en la fase ETL. Presentamos a continuación los datos disponibles.

El **DataFrame** se compone de tres ficheros de datos y un fichero de texto con el código en lenguaje M para la construcción de una tabla de calendario en el editor de consultas, nos centraremos en presentar los ficheros de datos que están en formato Excel.

 PBI Basic Calendar Table

 Superstore - Customer

 Superstore - Product

 Superstore - Sales

Ilustración 1

PBI Basic Calendar Table

Este es un fichero de texto que contiene el código en lenguaje M (estándar de Power Query) para la creación de una tabla calendario desde Power Query

Superstore – Customer

Es una tabla de dimensión donde identificamos a los clientes por su código (alfanumérico), nombre y tipo o segmento al que pertenece (consumidor individual, empresas y trabajador desde casa)

	A	B	C
1	Customer	Customer Name	Segment
2	AA-10315	Alex Avila	Consumer
3	AA-10375	Allen Arnold	Consumer
4	AA-10480	Andrew Allen	Consumer

Ilustración 2

Superstore – Sales

Esta es la tabla de hechos y que contiene la siguiente estructura

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Country	City	State	Postal Code	Region	Product ID	Sales	Quantity	Discount	Profit
2		1 CA-2016-152156	11/08/2016	11/11/2016	Second Class	CG-12520	United States	Henderson	Kentucky	42420	South	FUR-BO-10001798	261,96	2	0	41,914
3		2 CA-2016-152156	11/08/2016	11/11/2016	Second Class	CG-12520	United States	Henderson	Kentucky	42420	South	FUR-CH-10000454	731,94	3	0	219,58
4		3 CA-2016-138688	06/12/2016	6/16/2016	Second Class	DV-13045	United States	Los Angeles	California	90036	West	OFF-LA-10000240	14,62	2	0	6,8714
5		4 US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	United States	Fort Lauderdale	Florida	33311	South	FUR-TA-10000577	957,575	5	0,45	-383,03
6		5 US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335	United States	Fort Lauderdale	Florida	33311	South	OFF-ST-10000760	22,368	2	0,2	2,5164
7		6 CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710	United States	Los Angeles	California	90032	West	FUR-FU-10001487	48,86	7	0	14,169
8		7 CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710	United States	Los Angeles	California	90032	West	OFF-AR-10002833	7,28	4	0	1,9656
9		8 CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710	United States	Los Angeles	California	90032	West	TEC-PH-10002275	907,152	6	0,2	90,715
10		9 CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710	United States	Los Angeles	California	90032	West	OFF-BI-10003910	18,504	3	0,2	5,7825
11		10 CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710	United States	Los Angeles	California	90032	West	OFF-AP-10002892	114,9	5	0	34,47

	A	B	C	D	E	F	G	H
1	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Country	City
2		1 CA-2016-152156	11/08/2016	11/11/2016	Second Class	CG-12520	United States	Henderson
3		2 CA-2016-152156	11/08/2016	11/11/2016	Second Class	CG-12520	United States	Henderson

	I	J	K	L	M	N	O	P
1	State	Postal Code	Region	Product ID	Sales	Quantity	Discount	Profit
2	Kentucky	42420	South	FUR-BO-10001798	261,96	2	0	41,9136
3	Kentucky	42420	South	FUR-CH-10000454	731,94	3	0	219,582

Ilustración 3

En concreto los campos disponibles son:

Columna	Significado	Columna	Significado
Row ID	ID de fila	State	Estado
Order ID	ID de pedido	Postal Code	Código postal
Order Date	Fecha de pedido	Region	Región
Ship Date	Fecha de envío	Product ID	ID de producto

Ship Mode	Modo de envío	Sales	Ventas
Customer ID	ID de cliente	Quantity	Cantidad
Country	País	Discount	Descuento
City	Ciudad	Profit	Ganancia

Destacar que en estos datos de ventas el número de orden se repite porque pertenece a un mismo pedido de cliente que contiene varios artículos, (Ilustración 4).

	A	B	C	D	E	F
1	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID
2	1	CA-2016-152156	11/08/2016	11/11/2016	Second Class	CG-12520
3	2	CA-2016-152156	11/08/2016	11/11/2016	Second Class	CG-12520
4	3	CA-2016-138688	06/12/2016	6/16/2016	Second Class	DV-13045
5	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335
6	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335
7	6	CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710
8	7	CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710
9	8	CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710
10	9	CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710
11	10	CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710
12	11	CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710
13	12	CA-2014-115812	06/09/2014	6/14/2014	Standard Class	BH-11710
14	13	CA-2017-114412	4/15/2017	4/20/2017	Standard Class	AA-10480
15	14	CA-2016-161389	12/05/2016	12/10/2016	Standard Class	IM-15070
16	15	US-2015-118983	11/22/2015	11/26/2015	Standard Class	HP-14815

Ilustración 4

Otra cuestión especialmente relevante está relacionada con los campos fechas. En concreto nuestra empresa es del Reino Unido (Inglés – Estados Unidos) y por tanto su configuración de región es distinta de la nuestra y por tanto nos podemos encontrar con errores en la conversión, así debemos hacer los ajustes necesarios.

Este es un problema bastante común al trabajar con Power Query cuando los formatos de fecha no coinciden con la configuración regional del sistema.

Superstore – Product

Esta tabla de dimensión relacionada con los productos, disponiendo del código del producto (alfanumérico), de la categoría (Librerías, Tecnología, Material de Oficina, Muebles ..), subcategorías y nombre del producto.

	A	B	C	D
1	Product ID	Category	Sub-Category	Product Name
2	FUR-BO-10000112	Furniture	Bookcases	Bush Birmingham Collection Bookcase, Dark Cherry
3	FUR-BO-10000330	Furniture	Bookcases	Sauder Camden County Barrister Bookcase, Planked Cherry Finish
4	FUR-BO-10000362	Furniture	Bookcases	Sauder Inglewood Library Bookcases
5	FUR-BO-10000468	Furniture	Bookcases	O'Sullivan 2-Shelf Heavy-Duty Bookcases
6	FUR-BO-10000711	Furniture	Bookcases	Hon Metal Bookcases, Gray
7	FUR-BO-10000780	Furniture	Bookcases	O'Sullivan Plantations 2-Door Library in Landverly Oak

Ilustración 5

Se pide:

FASE I: ETL, Trabajando con Power Query

Priorizando el trabajo con el editor de consultas

Como hemos mencionado, es mejor transformar los datos antes de cargarlos al modelo para mejorar el rendimiento y la claridad y por ello en este caso la gran mayoría de transformaciones las vamos a hacer en PQ siempre que sea posible, la razón es que cuando trabajamos con gran cantidad de datos, para una mejor eficacia y rapidez aconsejamos trabajar con las consultas preparadas para después convertirla en DAX

El primer paso es crear nuestro libro de trabajo en Excel que llamaremos **Superstore UK** y accedemos al editor de consultas (PQ) donde comenzamos a realizar gran parte del trabajo, depuración y adaptación.

Conexión al Dataframe

Ajustes y depuración básicos

Comenzamos por **analizar el Dataframe** con especial atención al contenido de las columnas y el tipo de información que contiene, detectando posibles relaciones y tipos entre las fuentes de datos. Si existieran varias tablas o fuentes de datos, categorizarlas entre tablas de Dimensión o de Hechos.

Abrimos el editor de consultas, PQ y cargamos las tres tablas disponibles

- Tabla de Dimensión de Customers (Clientes)
- Tabla de Dimensión de Product (Productos)
- Tabla de Hechos de Sales (Ventas)

Creada la conexión realizamos el ajuste en el tipo de datos, como *usar la primera fila como encabezado*, *Cambiar Tipo*, la no existencias de errores, etc...

En este caso los ajustes realizados son muy básicos al no detectar errores, simplemente hemos promovido los encabezados y cambiado los tipos de campos a los más adecuado.

Sin embargo, destacar un problema con los campos fechas tanto de la columna Orden Date (fecha de pedido) como de Ship Date (fecha de entrega) de la consulta Sales que tiene un configuración regional diferente a la local debido a que la empresa está localizada en el Reino Unido y nosotros estamos operando con una sistema regional distinto el europeo y por tanto es necesario adaptarlo.

Consultas [3] = Table.PromoteHeaders(Sales_Sheet, [PromoteAllScalars=true])

ABC 123	Row ID	ABC 123	Order ID	ABC 123	Order Date	ABC 123	Ship Date
● Válido	100 %	● Válido	100 %	● Válido	100 %	● Válido	100 %
● Error	0 %	● Error	0 %	● Error	0 %	● Error	0 %
● Vacío	0 %	● Vacío	0 %	● Vacío	0 %	● Vacío	0 %
11		11	CA-2014-115812		06/09/2014		6/14/2014
12		12	CA-2014-115812		06/09/2014		6/14/2014
13		13	CA-2017-114412		4/15/2017		4/20/2017
14		14	CA-2016-161389		12/05/2016		12/10/2016
15		15	US-2015-118983		11/22/2015		11/26/2015
16		16	US-2015-118983		11/22/2015		11/26/2015
17		17	CA-2014-105893		11/11/2014		11/18/2014
18		18	CA-2014-167164		5/13/2014		5/15/2014
19		19	CA-2014-143336		8/27/2014		09/01/2014
20		20	CA-2014-143336		8/27/2014		09/01/2014
21		21	CA-2014-143336		8/27/2014		09/01/2014
22		22	CA-2016-137330		12/09/2016		12/13/2016
23		23	CA-2016-137330		12/09/2016		12/13/2016
24		24	US-2017-156909		7/16/2017		7/18/2017

Ilustración 6

Cambiar configuración regional de los campos fechas

El problema concreto con los campos fechas comentados consiste en que, en Europa, la configuración regional de la fecha más común es día/mes/año (DD/MM/AAAA), como en España, Francia, Alemania, etc. Sin embargo, en el Reino Unido, aunque geográficamente parte de Europa, la configuración regional es distinta y si nos fijamos esta es MM/DD/AAAA (mes/día/año), es decir la empresa tiene adoptado (inglés – Estados Unidos) y por tanto nos podemos encontrar con errores en la conversión, así debemos hacer los ajustes necesarios de conversión regional.

Este es un problema bastante común al trabajar con Power Query cuando los formatos de fecha no coinciden con la configuración regional del sistema. Para solucionarlo podríamos tomar en consideración lo siguiente:

Usar Locale al cambiar el tipo de datos

1. En Power Query, selecciona la columna de fecha.
2. Ve a **Transformar > Tipo de datos > Usar configuración regional**.
3. Elige **Tipo de datos: Fecha**.
4. En **Configuración regional**, seleccionamos **Inglés (Estados Unidos)** (o el país que corresponda al formato original).

Esto le indica a Power Query cómo interpretar correctamente las fechas antes de convertirlas al tipo de datos "Fecha", evitando errores por confusión entre día y mes.

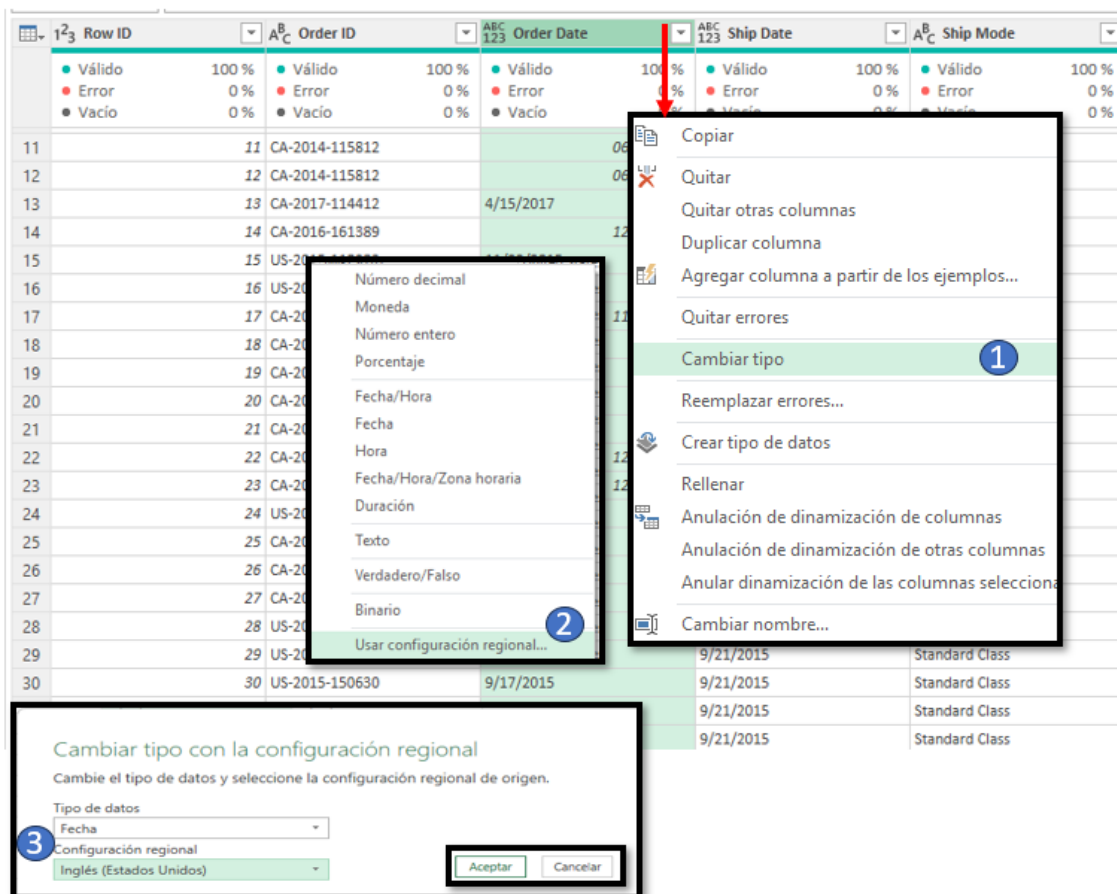
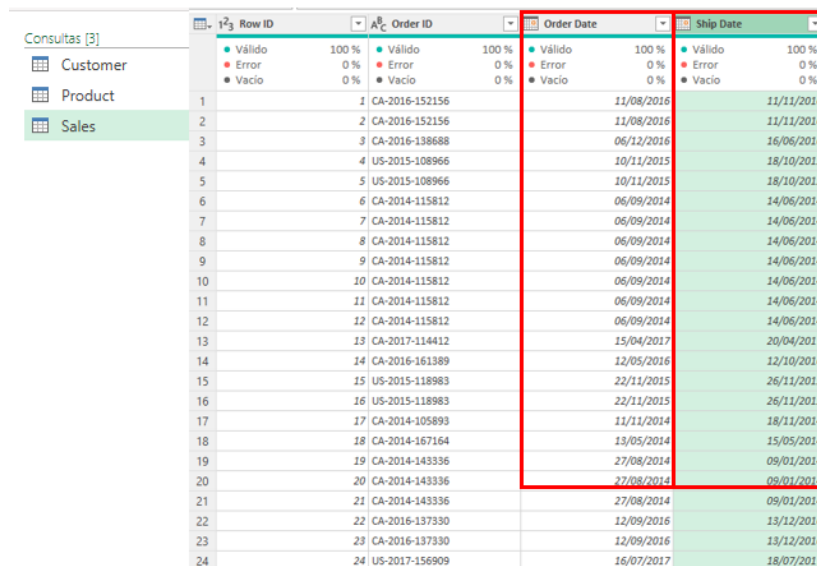


Ilustración 7

Esto lo hacemos para las dos columnas fechas señaladas y así el resultado de este ajuste es el siguiente:



Row ID	Order ID	Order Date	Ship Date
1	CA-2016-152156	11/08/2016	11/11/2016
2	CA-2016-152156	11/08/2016	11/11/2016
3	CA-2016-138688	06/12/2016	16/06/2016
4	US-2015-108966	10/11/2015	18/10/2015
5	US-2015-108966	10/11/2015	18/10/2015
6	CA-2014-115812	06/09/2014	14/06/2014
7	CA-2014-115812	06/09/2014	14/06/2014
8	CA-2014-115812	06/09/2014	14/06/2014
9	CA-2014-115812	06/09/2014	14/06/2014
10	CA-2014-115812	06/09/2014	14/06/2014
11	CA-2014-115812	06/09/2014	14/06/2014
12	CA-2014-115812	06/09/2014	14/06/2014
13	CA-2017-114412	15/04/2017	20/04/2017
14	CA-2016-161389	12/05/2016	12/10/2016
15	US-2015-118983	22/11/2015	26/11/2015
16	US-2015-118983	22/11/2015	26/11/2015
17	CA-2014-105893	11/11/2014	18/11/2014
18	CA-2014-167164	13/05/2014	15/05/2014
19	CA-2014-143336	27/08/2014	09/01/2014
20	CA-2014-143336	27/08/2014	09/01/2014
21	CA-2014-143336	27/08/2014	09/01/2014
22	CA-2016-137330	12/09/2016	13/12/2016
23	CA-2016-137330	12/09/2016	13/12/2016
24	US-2017-156909	16/07/2017	18/07/2017

Ilustración 8

Una vez creadas las conexiones, procedemos a modificarla para vincularla con un parámetro de conexión, por tanto, debemos crearlo.

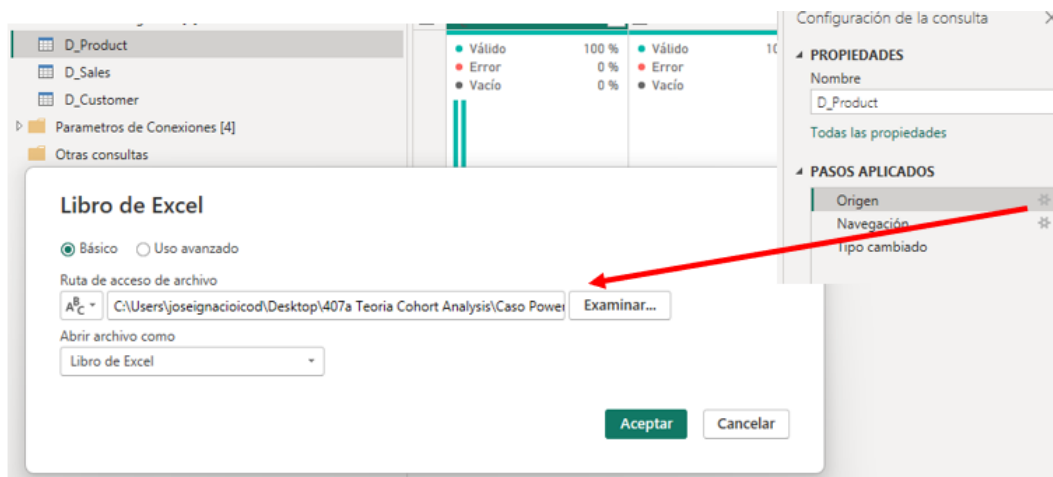


Ilustración 9

En esta primera fase es imprescindible llevar los ajustes necesarios en el editor de consultas (Power Query) como son el ajuste del tipo de columna, eliminar columna, verificar el tipo de datos y auditar los datos importados asegurándonos que no hay errores, ni valores en blanco en las columnas clave.

Es importante confirmar y revisar la no existencias de errores una vez llevada a cabo el proceso de ajuste y depuración contando con las opciones de Vista en Power Query que nos permite un análisis detallado columna por columna.

Creando parámetro de conexión general y para cada fichero

Procedemos a continuación a crear el parámetro de conexión que nos va a facilitar la actualización de las conexiones a las fuentes de datos en caso de modificaciones en la localización o cambios en los nombres y para ello vamos a crear dos tipos de conexión:

- **Conexión general** en la que se define la ruta a la carpeta donde está localizado las fuentes de datos.
- **Fichero**, en la que apunta al fichero específico al que conectaremos.

Estas dos variables van a definir los orígenes de conexión de cada consulta como presentamos a continuación.

Comenzamos con la conexión general para lo cual accedemos a la opción administración de parámetros y diseñamos la que hemos llamado ConexionCarpeta donde establecemos la ruta de acceso en el campo valor actual (Ilustración 10).

C:\Users\joseignacio\Superstore\Dataframe\

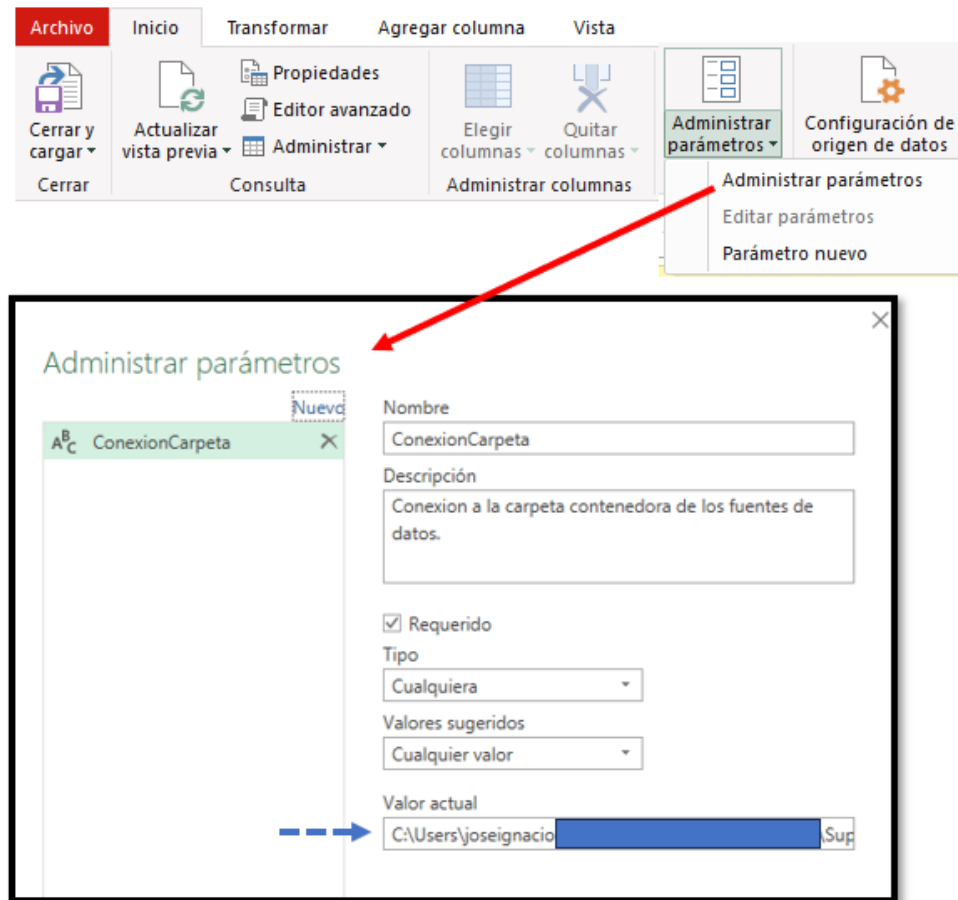


Ilustración 10

Procedemos a crear una conexión particular para cada fichero, donde solo especificamos el nombre del fichero y su extensión.

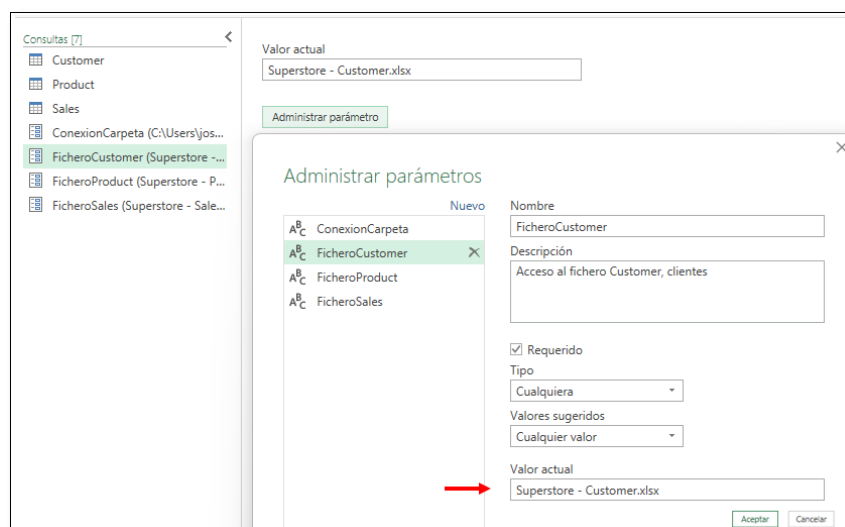


Ilustración 11

A continuación, revisaremos las conexiones creadas a las fuentes originales para adaptarlas a los parametros de conexión, así por ejemplo tendríamos para la conexión original al fichero Customer la siguiente mostrada en la Ilustración 12.

1. Seleccionamos la consulta (Customer) que queremos analizar y adaptar.
2. Accedemos al origen de la conexión
3. Nos muestra el cuadro de dialogo básico de configuración de la conexión, con la ruta de acceso al fichero
4. Seleccionamos la opción uso avanzado que nos permita configurar el parámetro de conexión de forma avanzada

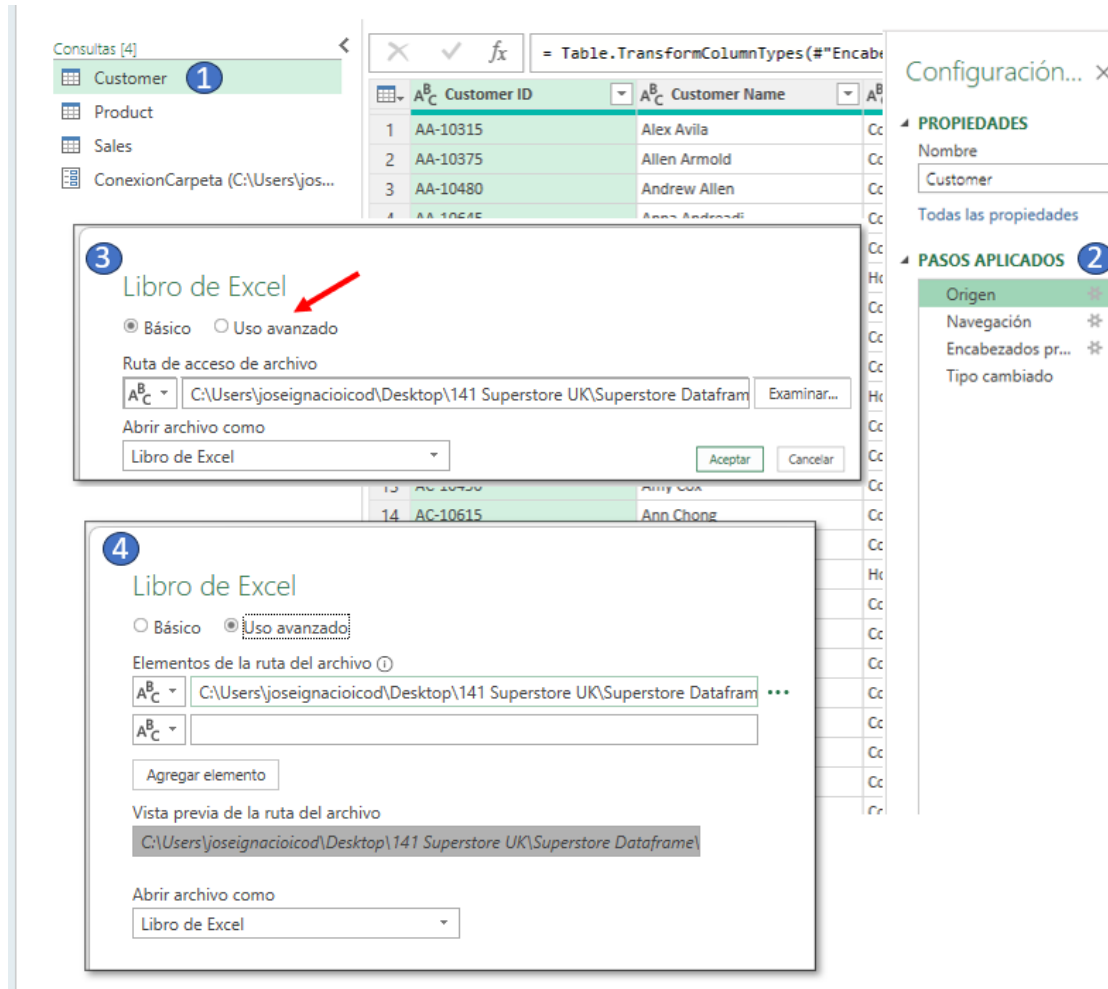


Ilustración 12

Adaptamos el parámetro conexión de la siguiente forma

Separamos la conexión en dos parametros, el de localización o ubicación (ConexionCarpeta) y apuntamos al nombre y extensión del fichero en este caso parámetro FicheroCustomer que contiene, tal y como se muestra en la Ilustración 11, al fichero Superstore – Customer.

Igualmente hacemos con el resto de ficheros.

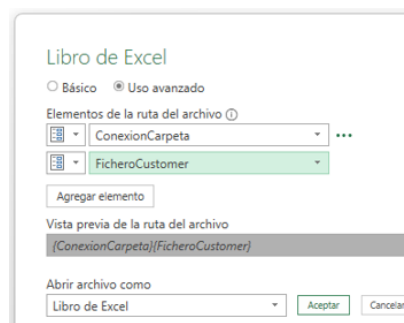


Ilustración 13

Así convertimos una conexión base (1) en una avanzada separando los dos parametros (2), ver Ilustración 14.

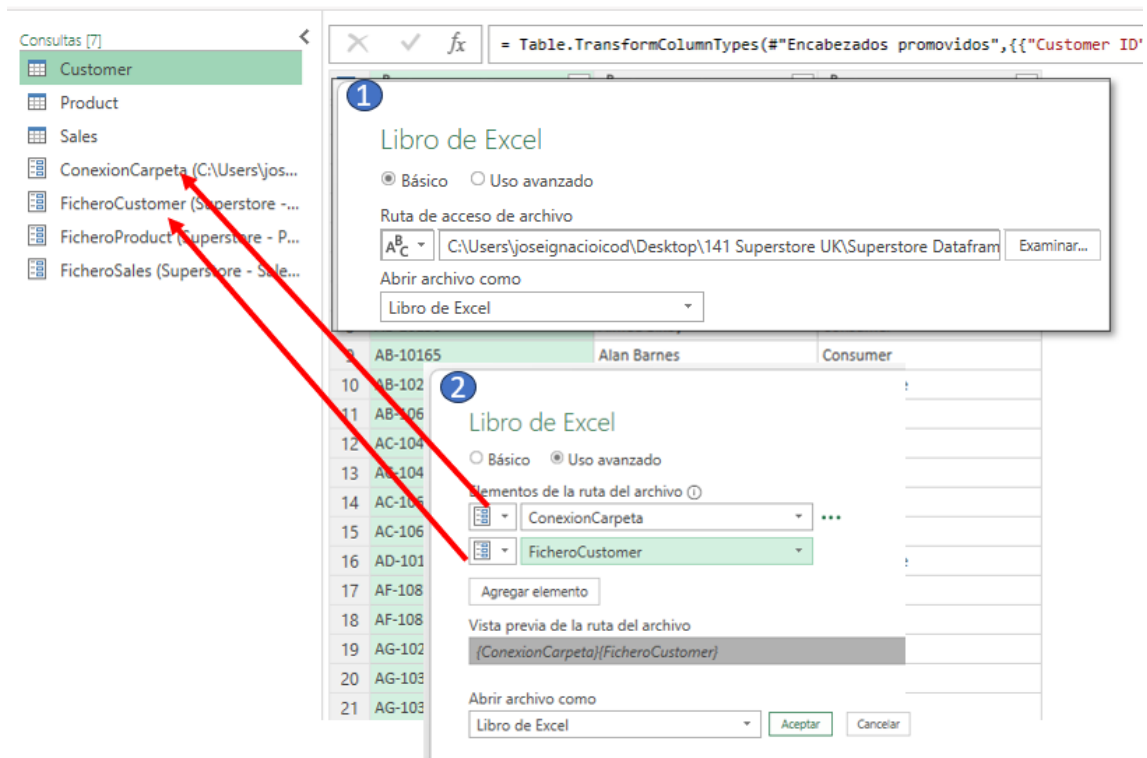


Ilustración 14

Organización de las consultas y parámetros

Tal y como se muestra en Ilustración 15 procedemos a organizar las consultas en grupos coherentes, en concreto en dos, Consultas Iniciales y la que tiene que ver con los Parámetros de Conexión.

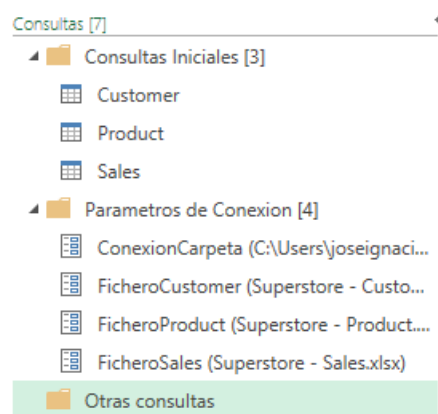


Ilustración 15

Configurar tabla calendario en el editor de consultas con lenguaje M

Aunque lo frecuente es crear la tabla de calendario con funciones DAX, también tenemos la opción más eficiente crearla en el editor de consultas con lenguaje M. En el anexo de este documento presentamos una comparativa entre las ventajas de usar funciones DAX o lenguaje M para crear una tabla calendario.

En este caso, como hemos señalado, disponemos de un fichero de texto PBI Basic Calendar Table (ver Ilustración 1) que contiene el código en lenguaje M que vamos a utilizar para crear nuestra tabla calendario en el editor de consulta Power Query.

Para ello generamos una consulta en blanco y activamos el Editor Avanzado de M donde pegamos el código, así los pasos secuenciales serían:

1. Copiar el código M del fichero
2. Crear una consulta en blanco
3. Activar el editor de consultas avanzados
4. Pegar el código.

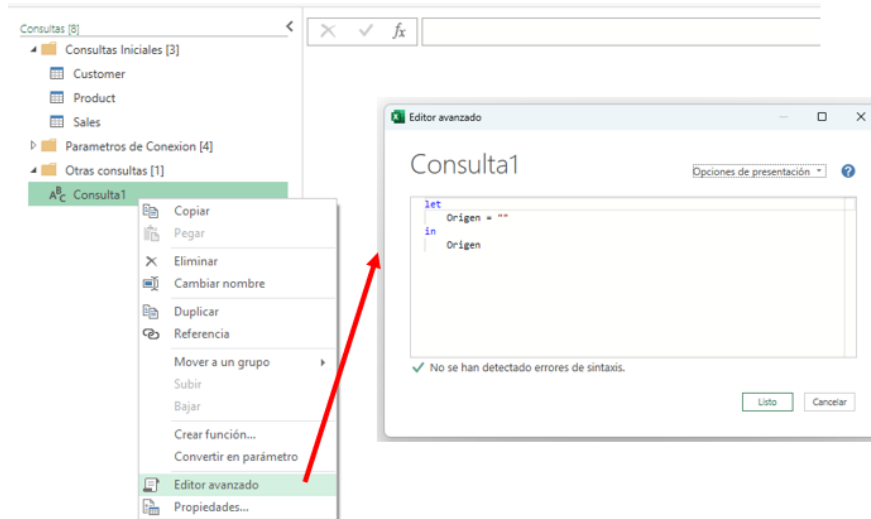


Ilustración 16

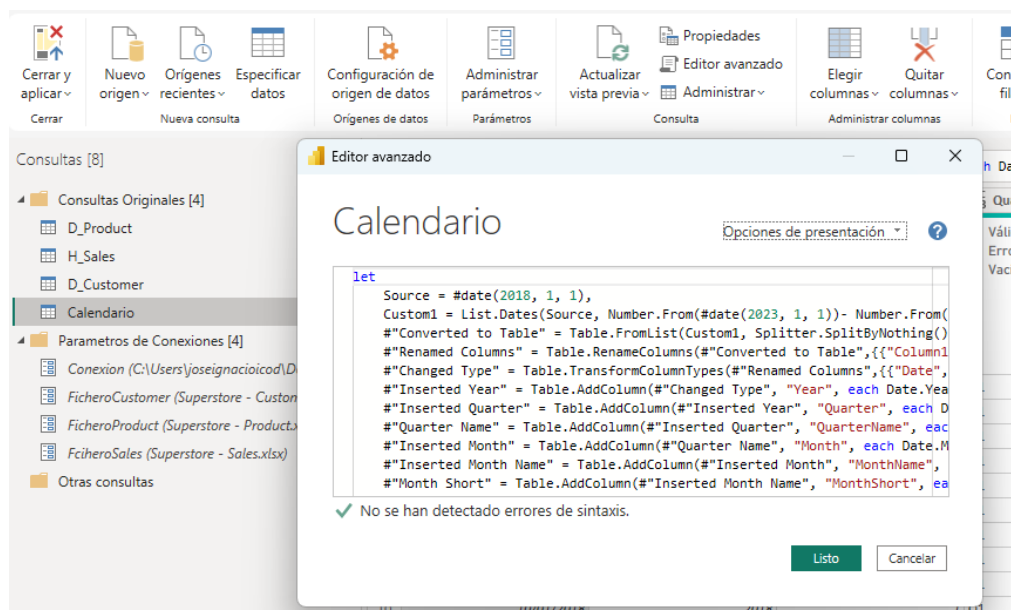


Ilustración 17

Una vez creada la debemos conectar al modelo de datos, recordando que **debemos marcarla como tabla calendario en el modelo de datos.**

Consultas [10]

Consultas Iniciales [6]

- Customer
- Product
- Sales
- Calendario

Parametros de Conexiones [4]

- Conexion (C:\Users\joseignacio\Documents\FicheroCustomer (Superstore - CustomerSales.xlsx))
- FicheroCustomer (Superstore - CustomerSales.xlsx)
- FicheroProduct (Superstore - ProductSales.xlsx)
- FicheroSales (Superstore - Sales.xlsx)

Otras consultas

Calendario

let Source = #date(2018, 1, 1), Custom1 = List.Dates(Source, Number.From(#date(2023, 1, 1)) - Number.From(#date(2018, 1, 1)), ...

Date	Year	Quarter	QuarterName
01/01/2018	2018	1	Q1
02/01/2018	2018	1	Q1
03/01/2018	2018	1	Q1
04/01/2018	2018	1	Q1

Month	MonthName	MonthShort	MonthYear	MonthYearSort	DayName
1	ene.	ene.	ene. 18	201801	lunes
1	ene.	ene.	ene. 18	201801	martes
1	ene.	ene.	ene. 18	201801	miércoles
1	ene.	ene.	ene. 18	201801	jueves

DayNameShort	DayOfWeek	WeekOfYear	StartOfWeek	EndOfWeek
lu.	2	1	30/12/2017	05/01/2018
ma.	3	1	30/12/2017	05/01/2018
mi.	4	1	30/12/2017	05/01/2018
ju.	5	1	30/12/2017	05/01/2018

Ilustración 18

Creando Consultas Referencia y ajustes

Introducción

En el anexo de este documento presentamos los conceptos básicos relacionados con las consultas de referencia, que es **una nueva consulta que se basa en la original**, y si se modifica la original, **la consulta de referencia se actualiza automáticamente**.

Su uso es ideal cuando se trabaja con modelos complejos y se necesita dividir el trabajo en partes mas manejables.

- Quieres reutilizar los datos procesados en una consulta para crear otra vista o aplicar diferentes transformaciones.
- Necesitas mantener una fuente de datos limpia y reutilizable sin alterar la original.
- Estás trabajando con modelos más complejos y necesitas dividir el trabajo en partes más manejables.

Esto lo hacemos porque la mayoría de las transformaciones necesarias para nuestro trabajo las vamos a realizar en Power Query sobre una copia (referenciadas) y no queremos modificar las consultas originales, además el trabajar todo lo posible como transformaciones, agregaciones en el editor de consultas es más eficiente y rápido que hacerlo con medidas o formulas DAX.

"RSales" como Referencia a la consulta H_Sales

Justificación

Comenzaremos por crear una consulta de referencia basada en la tabla de hechos H_Sales, la razón es que la citada tabla de hechos contiene todos los detalles de cada pedido de clientes, es decir un pedido puede contener diferentes articulos y por tanto el numero de pedido aparece duplicado y para un primer analisis que desarrollaremos no necesitamos ese nivel de detalle, necesitamos todos los pedidos agrupados por ID de Cliente (Customer ID) y numero de pedido (Order ID)

Es decir, como se señaló en la Ilustración 4 en los datos de ventas (sales) el número de orden del pedido (Order ID) se repite porque pertenece a un mismo pedido de cliente que contiene varios articulos, por tanto, vamos a proceder a agrupar esta consulta por pedidos, para trabajar con los datos consolidados por pedidos de clientes. Nos interesa disponer de una secuencia de pedidos de clientes para conocer el tiempo que llevan y por tanto no nos interesa tanto el detalle.

Primer paso, creacion de la consulta de referencia

Para la creacion la consulta de referencia de la tabla Sales, la seleccionamos y con el boton derecho accedemos a la opcion Referencia, que nos permitirá duplicarla, tal y como se muestra en la ilustracion adjunta, en este caso le cambiamos el nombre y la llamamos RSales, es decir mantenemos el nombre original añadiéndole R para identificarla como que es una consulta de referencia.

Sobre esta tabla será las que llevaremos a cabo los ajustes necesarios.

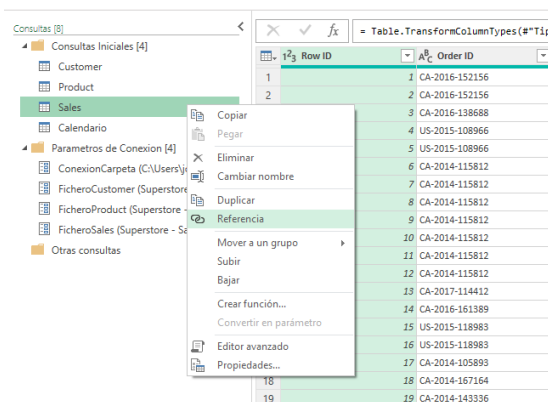


Ilustración 19

Segundo paso, creacion de ajustes, agrupación y ordenación

Sobre las tablas de referencia procedemos a llevar a cabo los ajustes necesarios y en este caso procederemos a los niveles de agrupamiento deseados, para lo cual seleccionamos la columna y con el boton derecho elegimos la opcion agrupar y vamos a la **opcion avanzado**, que nos da acceso a un cuadro de dialogo donde proponemos el siguiente orden de agrupamiento:

Customer ID (ID de cliente) - Order ID (ID de pedido) - Order Date (Fecha de pedido) - Región - Estado

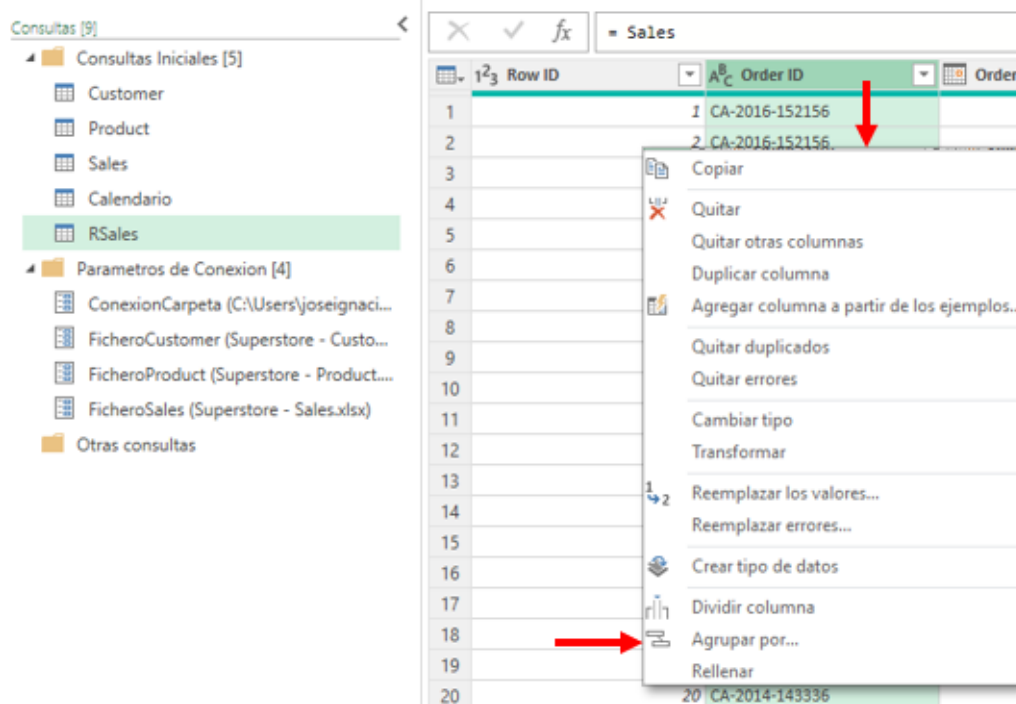


Ilustración 20

Señalar que, aunque bastaría solo con el código del pedido Order ID (ID de pedido) ya que queremos tener por cliente el número de pedidos no el detalle.

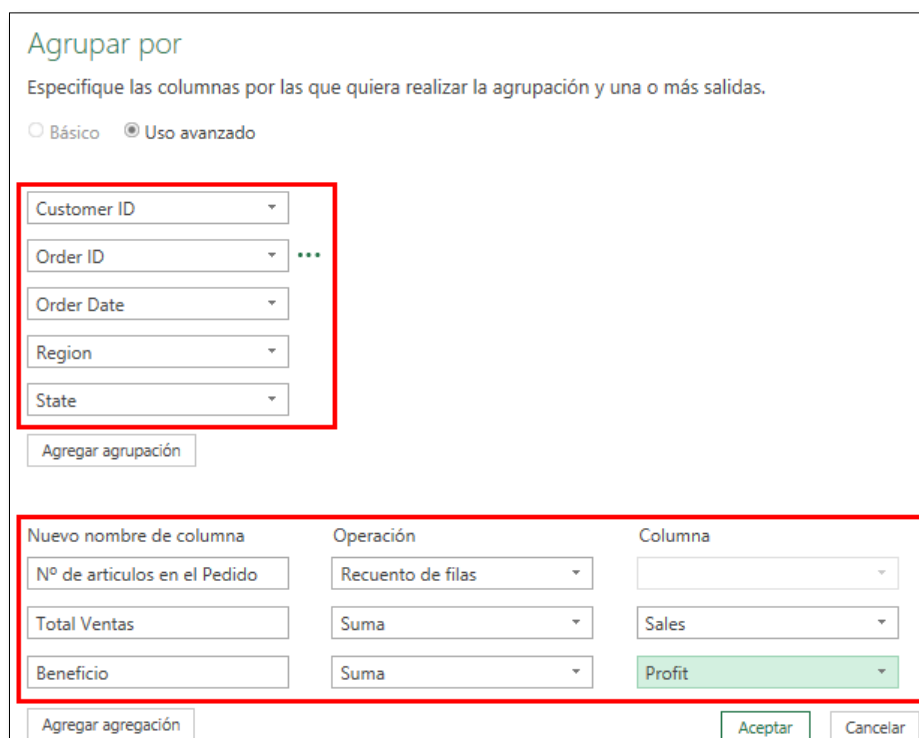


Ilustración 21

El resultado despues de agrupar y crear los nuevos campos seria:

Consultas [9]	AB _C Customer ID	AB _C Order ID	Order Date	AB _C Region	AB _C State	1 ₂ Nº de artículos en el Pedido	1.2 Total Ventas	1.2 Beneficio
1	CG-12520	CA-2016-152156	11/08/2016	South	Kentucky	2	993,9	261,4956
2	DV-13045	CA-2016-138688	26/12/2016	West	California	1	14,62	6,8714
3	SO-20335	US-2015-108966	10/11/2015	South	Florida	2	979,9455	-380,5146
4	BH-11710	CA-2014-115812	26/09/2014	West	California	7	3714,304	300,7687
5	AA-10480	CA-2017-114412	15/04/2017	South	North Carolina	1	15,552	5,4432
6	IM-15070	CA-2016-161389	12/05/2016	West	Washington	1	407,976	132,5922
7	HP-14815	US-2015-118983	22/11/2015	Central	Texas	2	71,354	-127,674
8	PK-19075	CA-2014-105893	11/11/2014	Central	Wisconsin	1	665,88	13,3176
9	AG-10270	CA-2014-167164	13/05/2014	West	Utah	1	55,5	9,99
10	ZD-21925	CA-2014-143336	27/08/2014	West	California	3	244,76	25,8774
11	KB-16585	CA-2016-137330	12/09/2016	Central	Nebraska	2	79,8	20,748
12	SF-20065	US-2017-156909	16/07/2017	East	Pennsylvania	1	71,372	-1,0196

Ilustración 22

En resumen, contamos con una consulta de referencia basada en la conexión a Superstore – Sales, agrupada por *Customer ID* (*ID de cliente*) - *Order ID* (*ID de pedido*) - *Order Date* (*Fecha de pedido*) – *Región* – *Estado* y ademas con tres columnas o campos calculados

Solo nos queda ordenar ascendentemente por cliente, es decir por *Customer ID* (*ID de cliente*) para lo cual nos situamos sobre la columna y seleccionamos con el boton derecho orden ascendente podemos ver que el primer cliente ordenado tiene 5 pedidos en distintas fechas (Ilustración 24).

Por tanto la configuración de la consulta consta de los tres pasos que se muestran en la Ilustración 23

Configuración de la consulta x

PROPIEDADES

Nombre
RSales

Todas las propiedades

PASOS APLICADOS

Origen
Filas agrupadas *

☒ Filas ordenadas

Ilustración 23

Consultas [9]	AB _C Customer ID	AB _C Order ID	Order Date
1	AA-10315	CA-2014-138100	15/09/2014
2	AA-10315		29/06/2017
3	AA-10315		31/03/2014
4	AA-10315		03/03/2016
5	AA-10315		10/04/2015
6	AA-10375		09/07/2017
7	AA-10375		02/03/2015
8	AA-10375		21/04/2014
9	AA-10375		07/10/2016
10	AA-10375		12/11/2017
11	AA-10375		05/08/2015
12	AA-10375		24/10/2014
13	AA-10375		14/11/2016
14	AA-10375		13/11/2015
15	AA-10480		05/04/2014

Ilustración 24

"RSales v2". Columnas primera fecha de compra, última fecha de compra y total de nº de ordenes.

Como evolución de la consulta anterior queremos proponer volver agrupar por codigo de cliente (*Customer ID*) y con unas nuevas columnas calculadas ademas de las ya disponibles, en concreto con columnas calculada de primera fecha de compra, última fecha de compra asi como numero de órdenes.

Por tanto, vamos a replicar la consulta RSales como hemos visto (Ilustración 19) y la llamaremos RSales v2 y procedemos a agrupar por los criterios definidos anteriormente pero ademas queremos tener acceso al resto de columnas disponible en la consulta origen (RSales), el resultado y la configuración se presenta en la Ilustración 25

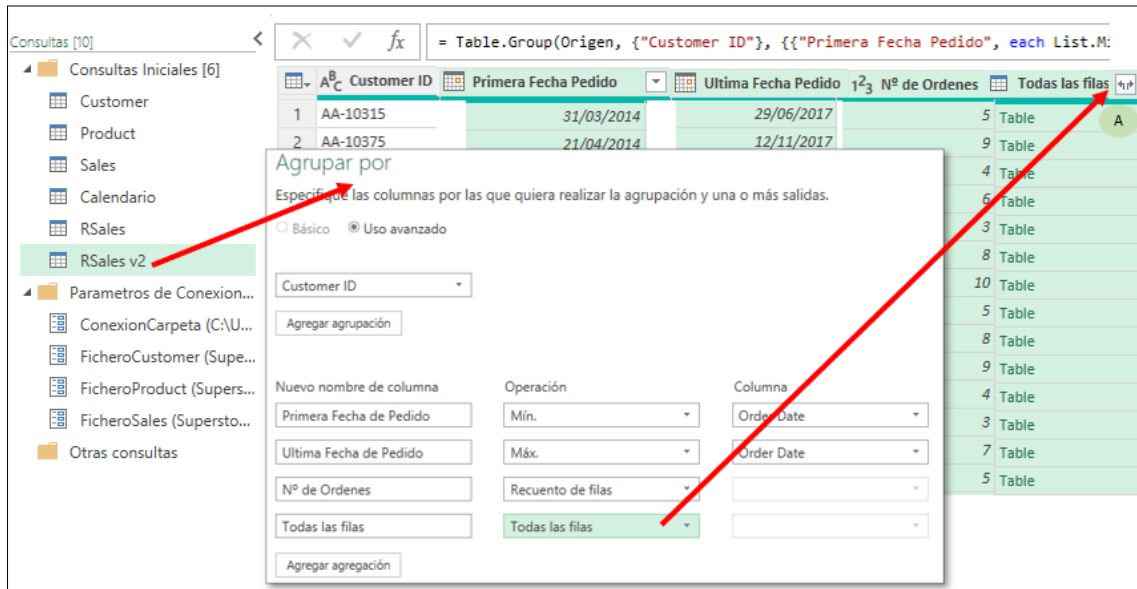


Ilustración 25

Si pulsamos sobre la columna Todas las Filas (A) se desplegará un cuadro de diálogo para seleccionar que columnas expandir de la consulta RSales además de las programadas.

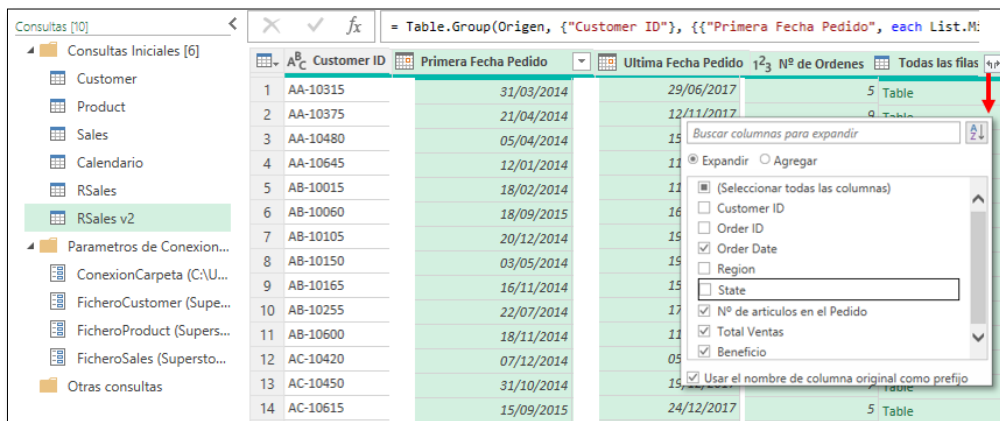


Ilustración 26

Si seleccionamos expandir, Fecha de la orden, Nº de artículos en el pedido, Total Ventas y Beneficio, el resultado de esta consulta Rsales v2, será:

Customer ID	Primera Fecha Pedido	Ultima Fecha Pedido	Nº de Ordenes
1 AA-10315	31/03/2014	29/06/2017	5
2 AA-10375	21/04/2014	12/11/2017	9
3 AA-10480	05/04/2014	12/11/2017	9
4 AA-10645	12/01/2014	12/11/2017	9
5 AB-10015	18/02/2014	12/11/2017	9
6 AB-10060	18/09/2015	12/11/2017	9
7 AB-10105	20/12/2014	12/11/2017	9
8 AB-10150	03/05/2014	12/11/2017	9
9 AB-10165	16/11/2014	12/11/2017	9
10 AB-10255	22/07/2014	12/11/2017	9
11 AB-10600	18/11/2014	12/11/2017	9
12 AC-10420	07/12/2014	12/11/2017	9
13 AC-10450	31/10/2014	12/11/2017	9
14 AC-10615	15/09/2015	24/12/2017	5

Ilustración 27

A la cual se le añaden las siguientes columnas

Customer ID	Order Date	Nº de artículos en el Pedido	Total Ventas	Total Beneficio
AA-10315	21/03/2014	2	726,548	267,4224
AA-10375	15/09/2014	2	29,5	13,2826
AA-10480	10/04/2015	1	26,96	7,0096
AA-10645	29/06/2017	2	374,48	96,505
AB-10015	03/03/2016	4	4406,072	747,1021
AB-10060	07/10/2016	1	16,768	1,4672
AB-10105	13/11/2015	1	84,96	6,372
AB-10150	05/08/2015	1	5,248	0,5904
AB-10165	02/03/2015	2	178,37	62,0858
AB-10255	09/07/2017	2	56,86	26,5552
AB-10600	14/11/2016	3	513,52	121,2536
AC-10420	24/10/2014	1	34,272	11,1384
AC-10450	21/04/2014	1	16,52	5,5755

Ilustración 28

Diseño del modelo de datos v1 – Modelados y relaciones

Agregar las consultas al modelo de datos

Las consultas creadas no queremos tenerlas cargadas en nuestra libro de trabajo por su tamaño o peso y en caso en que se haya cargado podemos eliminarlas, solo nos interesa que estén agradadas al modelo de datos para ello procedemos de la siguiente forma:

1. Confirmamos que nuestro libro de trabajo no tiene ninguna consulta cargada, en todo caso se borra las que existan.
2. Para cada conexión que queramos agregar al modelo de datos la seleccionamos y con el boton derecho accedemos a un nuevo cuadro de dialogo
3. En este cuadro de dialogo seleccionamos la opcion cargar que nos permitirá definir la disponibilidad de los datos de la consulta.
4. En nuestro caso nos interesa tener activo solo la conexión y no cargar los datos y activar el **"Agregar estos datos al modelo de datos"**

Este procedimiento lo haremos para todas las conexiones.

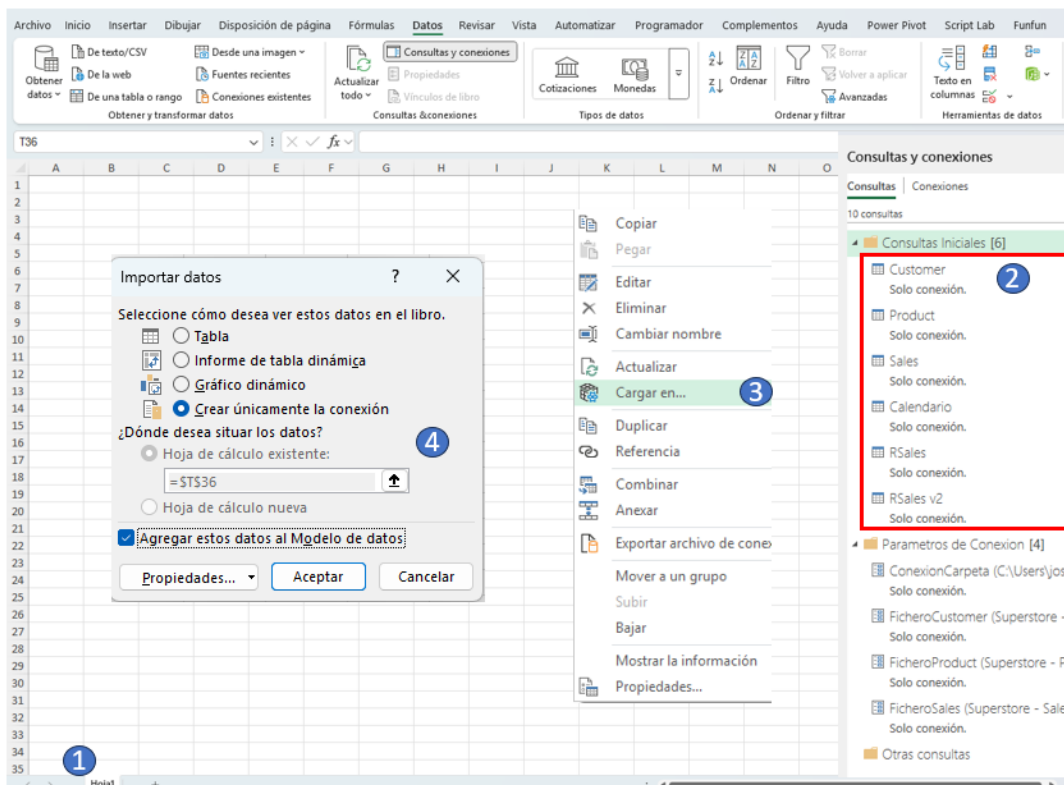


Ilustración 29

El resultado será que las seis consultas han sido cargadas al modelo de datos pero solo como conexión (Ilustración 30) ya que en el apartado 4 de la ilustración anterior (Ilustración 29) hemos seleccionado “Crear únicamente conexión” lo que implica que los datos no están almacenados en el libro de trabajo de Excel y por tanto no ocupa espacio pero esto implica que el usuario tiene acceso a la conexión o fuentes de datos, si por ejemplo se lleva este documento y lo transfiere a un lugar desde el cual no le permite acceder a las fuentes no podría trabajar con el libro, y en este caso necesitaría haber seleccionado la opción tabla para que con cada conexión se agregara una tabla por hoja y de esta forma si tendría acceso a los datos contenidos en el libro.

Ahora podemos acceder desde el fichero de trabajo a la opción Power Pivot – Administrar y observar como todos estos están agregados al modelo, aunque no conectados.

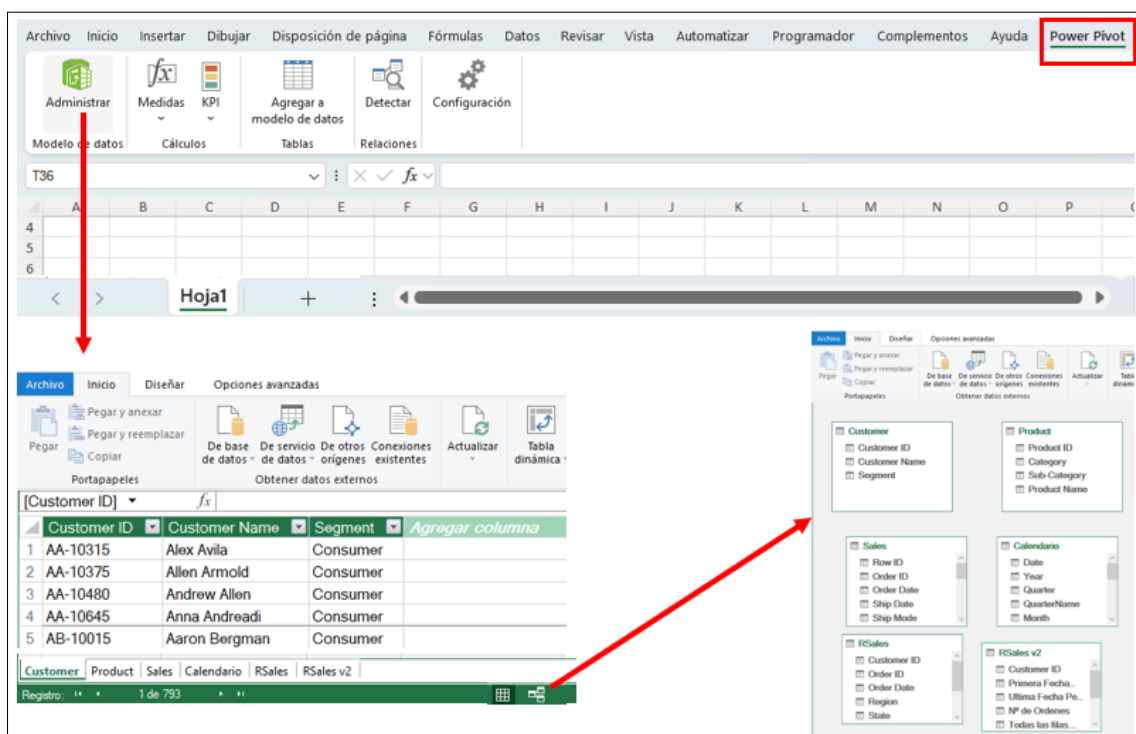
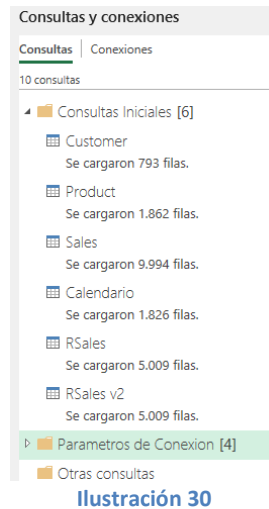


Ilustración 31

Crear tabla Calendario2 y marcar como tabla de fechas

Columnas base

Vamos a crear una segunda tabla de calendario, Calendario2 con DAX y con el contenido similar al propuesto en la Ilustración 18, simplemente para mostrar como también podemos crear este tipo de tablas desde DAX, por tanto, usaremos la función DAX CalendarAuto y procedemos en Power Pivot de la siguiente forma:

1. **Inicio > Vista de Diagrama.**
2. **Diseño > Tabla > Tabla de fechas > Nueva tabla.**
3. Escribe esta fórmula DAX: **Calendar = CALENDARAUQ**

Esto creará una tabla llamada Calendar con una columna de fechas que cubre automáticamente el rango de fechas detectado en el modelo de datos.

Podemos **agregar columnas adicionales (opcional)**, para enriquecer la tabla de fechas con columnas como año, mes, trimestre, etc. Estas columnas se agregan seleccionando la tabla Calendar, luego haciendo clic en **Agregar columna** y escribiendo la fórmula DAX correspondiente, ver anexo, *Ejemplo usando la función DAX CalendarAuto ()*.

El resultado requerido sería una tabla calendario similar con las siguientes columnas:

	Date	Año	Número de mes	Mes	MMM-AAAA	Número de día de la semana	Día de la semana
1	01/01/2014	1	1	enero	ene-2014	4	miércoles
2	02/01/2014	1	1	enero	ene-2014	5	jueves

Trimestre	DiaSemana	NombreDiaSemana	Mes-Año	AñoMes	SemanaDelAño
T1	miércoles	miércoles	4	201401	1
T1	jueves	jueves	5	201401	1

Ilustración 32

Nombre de la columna	Resultado	DAX
Date	01/01/2014	<code>CalendarAuto ()</code>
Año	2014	<code>YEAR([Date])</code>
Número de mes	1	<code>MONTH([Date])</code>
Mes	enero	<code>FORMAT([Date];"MMMM")</code>
MMM-AAAA	ene-14	<code>FORMAT([Date];"MMM-YYYY")</code>
Número de día de la semana	4	<code>WEEKDAY([Date])</code>
Día de la semana	miércoles	<code>FORMAT([Date];"DDDD")</code>
Cuatrimestre	Q1	<code>"Q" & FORMAT('Calendario2[Date]; "Q")</code>
Trimestre	T1	<code>"T" & ROUNDUP (MONTH (Calendario2[Date]) / 3;0)</code>
DiaSemana	4	<code>WEEKDAY(Calendario2[Date])</code>
NombreDiaSemana	miércoles	<code>FORMAT(Calendario2[Date];"DDDD")</code>
Mes-Año	ene-14	<code>FORMAT(Calendario2[Date];"mmm yy")</code>
AñoMes	201401	<code>YEAR(Calendario2[Date]) * 100 + MONTH(Calendario2[Date])</code>
SemanaDelAño	1	<code>=WEEKNUM([Date])</code>

Comienzo semana y fin de la semana.

Pero además necesitamos crear dos columnas especiales que nos devuelva la fecha de comienzo de la semana y la de finalización lo que nos permitirá realizar análisis con periodicidad semanal.

Para crear las dos columnas seguiremos la orientación dada por la consulta a la IA disponible en el anexo “*Crear columnas de fechas Fecha de Inicio y Fin de la Semana*”. En concreto los dos cálculos son:

ComienzoSemana	FinDeLaSemana
30/12/2013	05/01/2014
30/12/2013	05/01/2014

Ilustración 33

- ComienzoSemana: `=Calendario2[Date]-WEEKDAY(Calendario2[Date];2)+1`
- FinDeLaSemana: `= Calendario2[Date]-WEEKDAY(Calendario2[Date];2)+7`

Para finalizar, recordar que debemos **marcar esta tabla como tabla de fechas** por defecto.

Crear tabla Medidas

Tabla de Medidas, también deberemos disponer (crear) de una tabla para almacenar las medidas y que llamaremos “**Medidas**”

Modelo y relaciones

Toca finalmente definir el modelo de datos y establecer las relaciones, donde presentamos la siguiente propuesta:

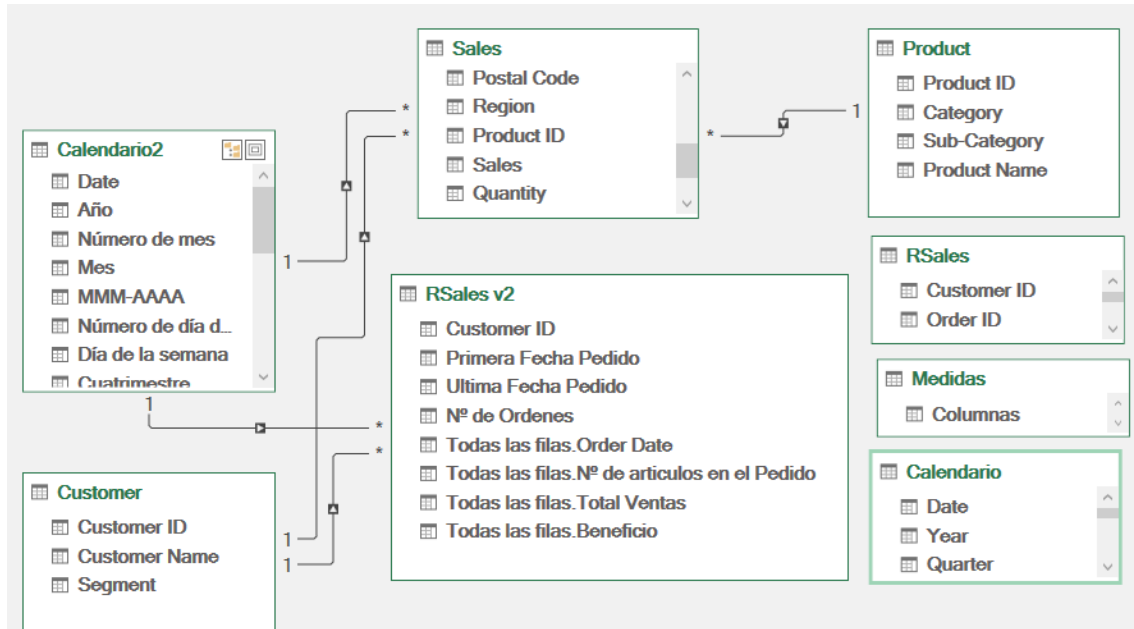


Ilustración 34

Administrar relaciones				
Crear Editar Eliminar				
Activo	Tabla 1	Cardinalidad	Dirección de filtro	Tabla 2
Sí	RSales v2 [Customer ID]	Varios a uno (*:1)	<< A RSales v2	Customer [Customer ID]
Sí	RSales v2 [Todas las filas.Order Date]	Varios a uno (*:1)	<< A RSales v2	Calendario2 [Date]
Sí	Sales [Customer ID]	Varios a uno (*:1)	<< A Sales	Customer [Customer ID]
Sí	Sales [Order Date]	Varios a uno (*:1)	<< A Sales	Calendario2 [Date]
Sí	Sales [Product ID]	Varios a uno (*:1)	<< A Sales	Product [Product ID]

Ilustración 35

FASE II: Informe I PDTE DE DESARROLLO

Orientación y notas

Anexo. Bibliografía y conceptos técnicos relacionados

Crear tabla calendario con DAX o con lenguaje M

Consulta IA: [En los modelos de datos que es mejor crear la... | Probar en Copilot Chat](#)

<https://chatgpt.com/share/68945d1a-8bc0-800a-be67-46bc96047a6f>

Tanto DAX como M (Power Query) pueden usarse para crear una tabla calendario, pero cada uno tiene ventajas según el contexto. Aquí te explico las diferencias para ayudarte a decidir cuál es mejor en tu caso:

Crear la tabla calendario con DAX

Ventajas:

- Se crea directamente en el modelo de datos, sin necesidad de cargar datos externos.
- Ideal si quieres una tabla dinámica que se adapte automáticamente al rango de fechas de tus datos.
- Puedes usar funciones como CALENDAR() o CALENDARAUTO() para generar fechas fácilmente.
- Muy útil si estás trabajando con modelos ya cargados y no quieres modificar el proceso de ETL.

Desventajas:

- Menos flexible para cálculos complejos o transformaciones previas.
- No puedes aplicar transformaciones de datos como en Power Query.

Crear la tabla calendario con M (Power Query)

Ventajas:

- Puedes aplicar lógica más compleja y transformar datos antes de que lleguen al modelo.
- Ideal si quieres una tabla calendario reutilizable y bien estructurada.

Desventajas:

- No se adapta automáticamente al rango de fechas de tus datos (a menos que lo programes).
- Requiere recargar el modelo si quieres cambiar el rango de fechas.

Tabla comparativa Calendario en DAX o en M

Aspecto	Power Query (M)	DAX
Rendimiento	✓ Mejor rendimiento. Se evalúa durante la carga, no en el modelo.	✗ Se calcula al abrir el modelo o actualizar, y se queda en memoria.
Carga	✓ Menos presión al motor de fórmula (VertiPaq).	✗ Añade carga al motor DAX.
Flexibilidad	✓ Puedes construir fechas desde tablas externas (como fechas de facturas).	⚠ Limitado: solo puedes usar funciones internas como CALENDAR() o CALENDARAUTO().

Transformaciones	☒ Puedes añadir columnas como "trimestre fiscal", "semana ISO", etc., de forma más controlada.	☒ Agregar columnas personalizadas en DAX puede volverse complejo y costoso en rendimiento.
Facilidad de depuración	☒ En Power Query puedes ver paso a paso qué falla.	☒ En DAX depurar errores puede ser más frustrante.
Automatización	☒ Más sencillo incorporar lógica dinámica de años desde datos reales.	☒ En DAX dependes de que ya haya datos cargados si usas CALENDARAUTO() (poco confiable).

¿Cuál elegir?

- Hay casos puntuales donde DAX puede tener sentido:
 - **Usa DAX** si necesitas rapidez, simplicidad y una tabla que se adapte automáticamente a tus datos.
 - **Modelos muy pequeños o personales**, donde el impacto en rendimiento es irrelevante. Montar un modelo rápido con CALENDARAUTO() y tirar.
- Pero en escenarios serios o profesionales, Power Query gana por goleada.
 - **Usa M** si necesitas más control, personalización o si estás trabajando en la etapa de transformación de datos.
 - Usa Power Query (M) para construir tu tabla calendario siempre que puedas. Más limpio, más potente y mejor para modelos escalables.

Ejemplo usando la función DAX CalendarAuto ()

Podemos usar la función DAX CalendarAuto para crear una tabla calendario y procedemos en Power Pivot de la siguiente forma:

4. **Inicio > Vista de Diagrama.**
5. **Diseño > Tabla > Tabla de fechas > Nueva tabla.**
6. Escribe esta fórmula DAX: **Calendar = CALENDARAUTO()**

Esto creará una tabla llamada Calendar con una columna de fechas que cubre automáticamente el rango de fechas detectado en el modelo de datos.

Podemos **agregar columnas adicionales (opcional)**, para enriquecer la tabla de fechas con columnas como año, mes, trimestre, etc. Estas columnas se agregan seleccionando la tabla Calendar, luego haciendo clic en **Agregar columna** y escribiendo la fórmula DAX correspondiente. Por ejemplo:

```

Año = YEAR('Calendar'[Date])
MesNombre = FORMAT('Calendar'[Date], "MMMM")
Mes-Año= FORMAT(Calendar [Date],"mmm yy")
Cuantrimestre= "Q" & FORMAT('Calendar'[Date], "Q")
"T" & ROUNDUP ( MONTH ( Calendario[Date] ) / 3;0 )
DiaSemana = WEEKDAY(Calendario[Date])
NombreDiaSemana:= FORMAT(Calendario[Date];"DDDD")
Año_mes=YEAR(Calendar[Date]) * 100 + MONTH(Calendar[Date])
SemanaDelAño= WEEKNUM([Date])
  
```

Ventajas de usar CALENDARAUTO()

- Automática: Detecta automáticamente el rango de fechas necesario según tus datos.
- Adaptable: Se ajusta automáticamente cuando agregues nuevos datos con fechas
- Eficiente: Crea solo las fechas necesarias basándose en tus datos
- Ideal para modelos dinámicos donde el rango de fechas puede cambiar.

las columnas a disponer básicas con serán:

Otra forma de código sería

```
Calendario =
ADDCOLUMNS(
    CALENDARAUTO(),
    "Año", YEAR([Date]),
    "Trimestre", "Q" & QUARTER([Date]),
    "Mes", FORMAT([Date], "mmm"),
    "Número Mes", MONTH([Date])
)
```

Crear columnas de fechas Fecha de Inicio y Fin de la Semana

Estas columnas nos permiten conocer cuando empieza y finaliza la semana lo que nos permite realizar análisis con una perspectiva temporal semanal.

Consulta IA. [En Excel trabajando con Power Pivot sobre una... | Probar en Copilot Chat](#)

Para obtener las columnas **StartOfWeek** y **EndOfWeek** en una tabla calendario en Power Pivot (usando DAX), puedes crear columnas calculadas que determinen el inicio y el fin de la semana para cada fecha. Aquí te muestro cómo hacerlo:

- **1. StartOfWeek (Inicio de semana)**

StartOfWeek = 'Calendario'[Fecha] - WEEKDAY('Calendario'[Fecha], 2) + 1

- WEEKDAY([Fecha], 2) devuelve el día de la semana (1 = lunes, 7 = domingo).
- Restamos ese valor menos 1 para llegar al lunes de esa semana.

- **2. EndOfWeek (Fin de semana)**

EndOfWeek = 'Calendario'[Fecha] - WEEKDAY('Calendario'[Fecha], 2) + 7

- Similar al anterior, pero sumamos 7 para llegar al domingo de esa semana.
- **Notas:**
- Estas fórmulas asumen que la semana comienza el **lunes**. Si prefieres que comience el **domingo**, cambia el segundo argumento de WEEKDAY a 1.
- Asegúrate de que la columna [Fecha] esté en formato de fecha y que la tabla calendario esté correctamente relacionada con tus datos.

Consultas de referencia

Concepto

En Power Query, una **consulta de referencia** es una **consulta nueva que se basa en otra existente**, como si fuera una copia *vinculada* a la original. Pero no es una copia literal: **es una vista dinámica**, lo que significa que si la consulta original cambia, la de referencia también se actualiza automáticamente. Esto se hace utilizando la opción **"Referencia"** en el editor de Power Query.

Es decir:

- La nueva consulta **hereda los resultados** de la consulta original.
- **No copia los pasos** de transformación, solo toma el resultado final como punto de partida.
- Si modificas la consulta original, **la consulta de referencia se actualiza automáticamente**.

Usos y ventajas

Las consultas de referencia tienen varias utilidades importantes:

- *Reutilización de lógica*

Puedes crear una consulta base con todos los pasos de transformación necesarios (filtrado, limpieza, etc.) y luego crear consultas de referencia que usen esa base para aplicar transformaciones adicionales sin modificar la original.

- *Modularidad*

Permite dividir el proceso de transformación en partes más manejables. Por ejemplo, puedes tener una consulta que limpia los datos y otra que los agrupa, sin tener que repetir pasos.

- *Evitar duplicación*

En lugar de copiar y pegar pasos entre consultas, puedes referenciar una consulta existente y construir sobre ella, lo que facilita el mantenimiento.

- *Creación de múltiples vistas*

Puedes generar diferentes versiones de los datos (por ejemplo, por región, por producto, por fecha) a partir de una misma consulta base.

- *Crear variantes de una base común*

Imagina que partes de una misma tabla y necesitas:

- Una versión solo con productos de España.
- Otra solo con productos de Latinoamérica.

En lugar de duplicar la consulta base, haces dos referencias, y a cada una le aplicas sus propios filtros.

Cuidado: no es lo mismo que "duplicar"

- **Duplicar:** crea una copia *totalmente independiente* de la consulta, con sus propios pasos. No se actualiza si la original cambia.
- **Referencia:** crea una consulta que depende de la original.

En resumen

Acción	¿Independiente?	¿Se actualiza con cambios?	¿Para qué sirve?
Duplicar	Sí	No	Hacer pruebas o versiones completamente separadas.
Referenciar	No	Sí	Reutilizar, modularizar, y mantener consistencia.

Ejemplos

Supongamos que tienes una consulta llamada VentasLimpias que ya ha filtrado y limpiado los datos de ventas. Si quieres crear una vista solo con las ventas de 2025, puedes hacer una consulta de referencia sobre VentasLimpias y aplicar un filtro adicional por año

Otro ejemplo sería *Control de inventario por categorías*, supón que tienes una consulta llamada InventarioGeneral que contiene todos los productos de una tienda, con columnas como: Producto, Categoría, Stock y Precio

Quieres crear **informes separados** por categoría (por ejemplo, Electrónica, Ropa, Hogar), pero **sin duplicar la lógica de limpieza** que ya aplicaste en InventarioGeneral.

Paso a paso:

1. **Consulta base:** InventarioGeneral ya tiene los datos limpios y estructurados.

2. **Consulta de referencia:** Creas una nueva consulta llamada InventarioElectronica haciendo click derecho sobre InventarioGeneral → "**Referencia**".
3. En InventarioElectrónica, aplicas un filtro: Categoría = "Electrónica".
4. Repites el proceso para otras categorías (InventarioRopa, InventarioHogar, etc.).

Las ventajas serian

- Si cambias algo en InventarioGeneral (por ejemplo, mejoras la limpieza de datos), **todas las consultas de referencia se actualizan automáticamente**.
- Evitas repetir pasos en cada consulta.
- Puedes crear **informes dinámicos** por categoría sin perder consistencia.

Otro ejemplo seria, que tenemos una consulta llamada Ventas2025 que filtra y transforma datos de ventas. Si hacemos una referencia a Ventas2025, podemos crear una nueva consulta llamada VentasPorRegión que agrupe los datos por región, sin modificar la lógica original de Ventas2025.

Esto lo hacemos porque la mayoría de las transformaciones necesarias para nuestro trabajo las vamos a realizar en Power Query sobre una copia (referenciadas) y no queremos modificar las consultas originales, además el trabajar todo lo posible como transformaciones, agregaciones en el editor de consultas es más eficiente y rápido que hacerlo con medidas o formulas DAX.