Power BI

GUÍA PARA CREAR UN CALENDARIO EN POWER BI

CONTENIDOS

La importancia de crear un calendario _____ 3

Cómo crear un calendario en una tabla de Power BI _____ 4

1. Importar datos en Power BI _____ 4
2. Crear el calendario en una tabla con CALENDARAUTO _____ 4

Completar el calendario con funciones DAX _____ 6

3. Añadir una columna con el nombre del mes con FORMAT _____ 6
4. Añadir una columna con el número del mes con MONTH _____ 7
5. Añadir una columna con el día de la semana con FORMAT _____ 9
6. Añadir una columna con el número del día de la semana con WEEKDAY _____ 9
7. Añadir una columna para el número de la semana con WEEKNUM _____ 10
8. Añadir una columna para el trimestre con QUARTER _____ 11
9. Añadir una columna para el semestre con ROUNDUP Y DIVIDE _____ 11
10. Añadir una columna para el año con YEAR _____ 13
11. Añadir una columna para el día del año con DATEDIFF _____ 14

Relacionar la tabla Calendario con la Tabla Maestra _____ 16

Realizar todo con una sola fórmula DAX _____ 17

LA IMPORTANCIA DE CREAR UN CALENDARIO

Cuando trabajamos en Power BI es imprescindible que una de nuestras tablas sea un calendario. De hecho**, es aconsejable que uno de los primeros pasos cuando iniciamos un estudio con esta herramienta sea crear una tabla calendario**. Esto nos va a permitir trabajar con nuestros datos teniendo en cuenta un eje temporal, y esto es un aspecto básico a la hora de hacer un análisis de datos.

Estos son dos ejemplos comunes en los que es clave tener una tabla calendario:

- Queremos estudiar la evolución de las ventas a lo largo de los años.
- Queremos hacer un estudio sobre la diferencia de ingresos entre el mes actual y el mes pasado.

Por lo tanto, vemos que **tener un calendario dentro de nuestro informe en una tabla independiente es un pilar fundamental que nos permite realizar el análisis.** Pero no solo debemos crearlo, sino que también debemos modelarlo con la finalidad de que nos quede un calendario formado por columnas que contengan campos apropiados para el estudio que queremos realizar.

Así pues, vamos a ver paso a paso **cómo crear un calendario en una tabla** y cómo transformarlo para que tenga la máxima utilidad posible. Todo esto lo haremos mediante funciones DAX muy sencillas que vamos a ir incorporando para que se puedan utilizar en cualquier informe.

CÓMO CREAR UN CALENDARIO EN UNA TABLA DE POWER BI

1. Importar datos en Power BI

Este es el primer paso para realizar cualquier informe en Power BI. Más allá de crear un calendario, lo primero que se debe hacer si vamos a realizar un análisis de datos es importar estos datos en nuestra herramienta. Aunque parezca un paso obvio, es muy necesario tener en cuenta un detalle: si queremos realizar un calendario, **el conjunto de datos que tenemos que importar debe contener obligatoriamente un campo que sea de fecha.** Si esto no es así, cuando utilicemos la función DAX para generar un calendario, Power BI no tendrá ningún campo referente para crearlo.

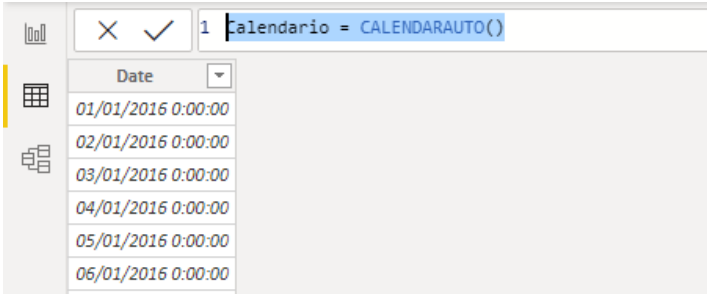
2. Crear el calendario en una tabla con CALENDARAUTO

Crear la tabla

Para generar el calendario, debemos crear una tabla calculada e introducir la función CALENDARAUTO. Si quieres saber más sobre cómo crear una tabla calculada puedes leerlo aquí.

Con lo cual, la función DAX que tendremos que introducir en la tabla va a ser la siguiente:

```
Calendario =
CALENDARAUTO ()
```



Como vemos, con esta función se genera automáticamente una tabla independiente formada por una columna que detecta la primera y última fecha de nuestro conjunto de datos del modelo cargado. Aunque ahora mismo tengamos una tabla básica, realmente ya podemos decir que hemos creado un calendario en Power BI.

Aplicar formato

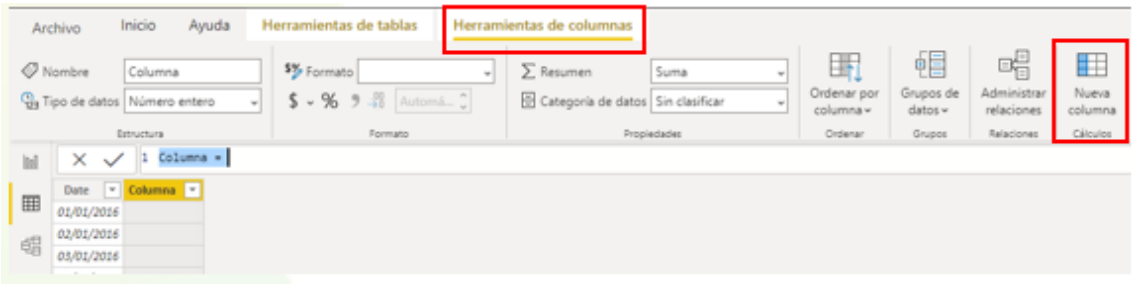
Una vez ya la tenemos creada, podemos dar formato a esta columna. En nuestro caso vemos que también nos aparece la hora, pero supongamos que queremos eliminarla en cada una de las filas, o darle otro formato específico. Para ello seleccionaremos la columna e iremos a la pestaña ***Herramientas de Columnas,*** y dentro de la sección ***Formato*** seleccionaremos el tipo de formato que queremos del desplegable.

COMPLETAR EL CALENDARIO CON FUNCIONES DAX

Como hemos visto, ya tenemos creado nuestro calendario, pero esa es solo la primera parte. Nuestro calendario debe contener un conjunto de columnas que serán los campos que vamos a necesitar a lo largo de nuestro análisis para que nos sean fáciles de interpretar cada una de las visualizaciones.

Por lo tanto, el siguiente paso en la creación de un calendario consiste en ir creando nuevas columnas a partir de la inicial, utilizando funciones DAX.

Para crear una columna en la cual poder introducir nuestras funciones tenemos que ir a la pestaña de ***Herramientas de Columnas,*** y en la sección de ***Cálculos*** seleccionar la opción **"Nueva Columna".**

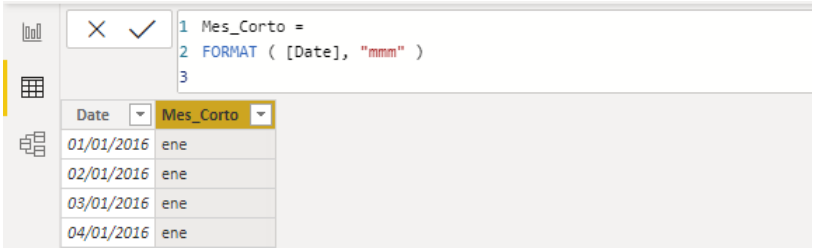


3. Añadir una columna con el nombre del mes con FORMAT

La segunda columna de nuestro calendario va a ser **el nombre del mes**. Esto lo haremos mediante la función **FORMAT**. De hecho, configuraremos la función para que solo nos salgan **las tres primeras letras del mes**. Con lo cual, en vez de tener "noviembre" se nos indicará el mes solo con "nov".

Para ello vamos a utilizar la siguiente función DAX:

```
Mes_Corto =
FORMAT ( [Date], "mmm" )
```



El único “problema” de esta función es que Power BI lo interpreta como “texto” y al crear una visualización con este nuevo campo, se nos ordenarían los meses por orden alfabético, es decir, nos quedaría ordenado de tal forma:

Mes_Corto
abr
ago
dic
ene
feb
jul
jun
mar
may
nov
oct
sep

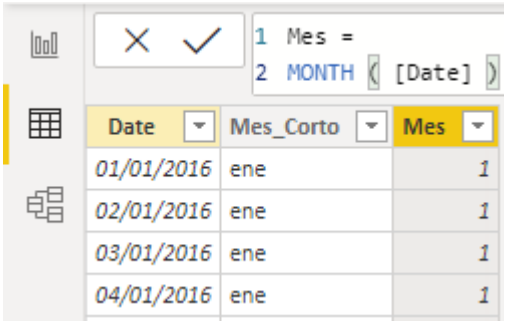
Como vemos, esto no tiene ningún sentido, ya que los informes se harían complicados de entender. Precisamente para ordenar este nuevo campo de forma correcta, necesitamos realizar otra columna usando la función MONTH, que explicamos en el siguiente punto.

4. Añadir una columna con el número del mes con MONTH

La siguiente columna en nuestra tabla va a ser el **mes en formato de número**. Es decir, cuando nuestra fila haga referencia a enero, esta columna va a contener un 1; cuando haga referencia a febrero, va a contener un 2; y así sucesivamente. Como ya hemos dicho, esto también nos va a ser útil para **ordenar correctamente la columna anterior.**

Para crear esta columna vamos a usar la siguiente función DAX:

```
Mes =
MONTH ( [Date] )
```

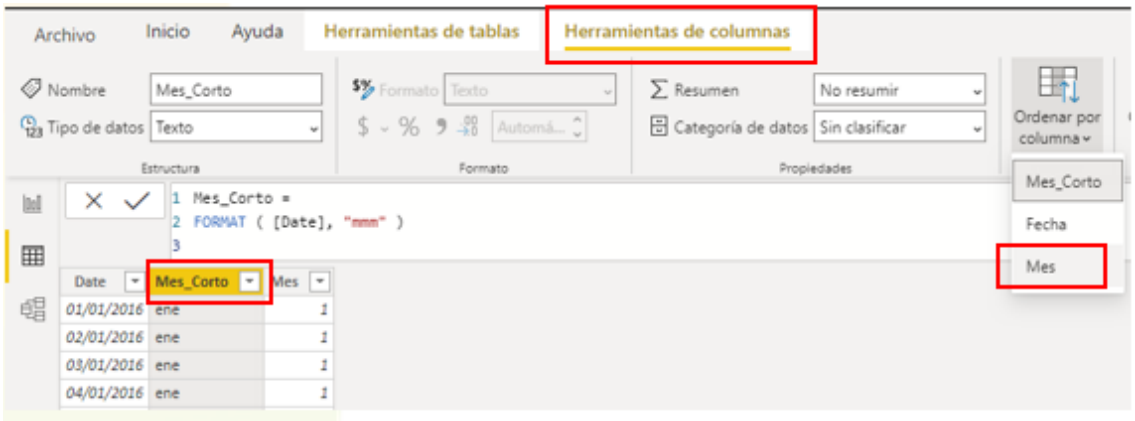


Como vemos, ahora a cada mes se le ha asignado su número de mes.

Ordenar la columna “Mes_Corto” correctamente

Volviendo al problema que teníamos relacionado con el orden de la columna “Mes_Corto”, lo que debemos hacer es decirle a Power BI que la columna “Mes_Corto” de mi tabla me la ordene teniendo en cuenta los números del mes. Este paso es muy sencillo y no requiere ninguna función DAX.

Para ello, debemos seleccionar la columna “Mes_Corto”, y en la pestaña de ***Herramientas de columnas***, hacer clic en **Ordenar por columna** seleccionando la columna “Mes”.



De esta forma, si ahora creamos una visualización con la columna de “Mes_Corto”, esta ya estará ordenada correctamente. Veámoslo con una tabla sencilla como hemos hecho antes:

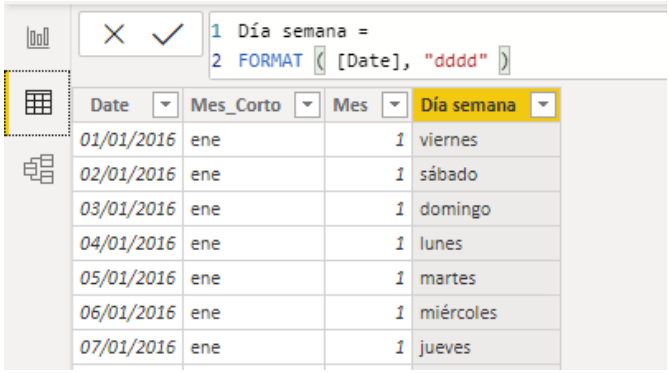
Si quieres saber más sobre la función Ordenar por columna, puedes leer nuestro post **“Cómo crear un orden personalizado en Power BI”.**

5. Añadir una columna con el día de la semana con FORMAT

Aplicamos la misma fórmula **FORMAT** que hemos utilizado en "Mes_Corto", pero en este caso vamos a utilizar la función para crear una columna que contenga **el día de la semana.**

La función DAX es la siguiente:

```
Día semana =
FORMAT ( [Date], "dddd" )
```



Date	Mes_Corto	Mes	Día semana
01/01/2016	ene	1	viernes
02/01/2016	ene	1	sábado
03/01/2016	ene	1	domingo
04/01/2016	ene	1	lunes
05/01/2016	ene	1	martes
06/01/2016	ene	1	miércoles
07/01/2016	ene	1	jueves

6. Añadir una columna con el número del día de la semana con WEEKDAY

Exactamente de la misma forma que hemos ordenado correctamente el mes, vamos a hacerlo con los días de la semana. Para ello, como ya sabemos, necesitamos crear otra columna que contenga el **número del día de la semana**. Es decir, si la fila hace referencia al lunes, esta columna va a contener el número 1; si la fila hace referencia al martes, va a contener el 2; y así sucesivamente. En este caso vamos a usar la función **WEEKDAY**.

Para realizar esto vamos a usar la siguiente función DAX:

```
Día Semana Núm =
WEEKDAY ( [Date], 2 )
```

En este caso, vemos que dentro de la función hemos añadido **el número 2. Esto es debido a que, por defecto, Power BI, interpreta que la semana empieza el domingo y termina el lunes**. Para cambiar eso y hacer que nuestras semanas empiecen el lunes y terminen el domingo, hemos añadido el número 2.

```
1 Día Semana Núm =
2 WEEKDAY ( [Date], 2 )
```

Date	Mes_Corto	Mes	Día semana	Día Semana Núm
01/01/2016	ene	1	viernes	5
02/01/2016	ene	1	sábado	6
03/01/2016	ene	1	domingo	7
04/01/2016	ene	1	lunes	1
05/01/2016	ene	1	martes	2
06/01/2016	ene	1	miércoles	3
07/01/2016	ene	1	jueves	4
08/01/2016	ene	1	viernes	5
09/01/2016	ene	1	sábado	6

A continuación, solo quedaría ordenar la columna "Día semana" por la columna "Día Semana Núm". Para ello, tenemos que usar la opción **Ordenar por columna** hemos visto con anterioridad al ordenar los meses de la columna "Mes_Corto".

7. Añadir una columna para el número de la semana con WEEKNUM

En la siguiente columna vamos a introducir **el número de semana de la fecha concreta de cada fila**. Es decir, vamos a tener una columna numérica que nos dirá en qué semana del año nos encontramos.

Para ello vamos a utilizar la siguiente función DAX:

```
Núm de Semana =
WEEKNUM ( [Date], 2 )
```

```
1 Núm de Semana =
2 WEEKNUM ( [Date], 2 )
```

Date	Mes_Corto	Mes	Día semana	Día Semana Núm	Núm de Semana
01/01/2016	ene	1	viernes	5	1
02/01/2016	ene	1	sábado	6	1
03/01/2016	ene	1	domingo	7	1
04/01/2016	ene	1	lunes	1	2
05/01/2016	ene	1	martes	2	2
06/01/2016	ene	1	miércoles	3	2
07/01/2016	ene	1	jueves	4	2
08/01/2016	ene	1	viernes	5	2

Volvemos a indicar en el segundo argumento que la semana empieza en lunes gracias al "2".

Una vez se nos ha creado la columna, ya podemos ver, por ejemplo, que las tres primeras filas hacen referencia a la primera semana del año, y así con cada una de las filas de nuestra tabla.

8. Añadir una columna para el trimestre con QUARTER

Esta nueva columna nos va a mostrar el **trimestre en el que nos encontramos en cada una de las filas de la tabla**. Para ello, la función principal que vamos a usar es **QUARTER**.

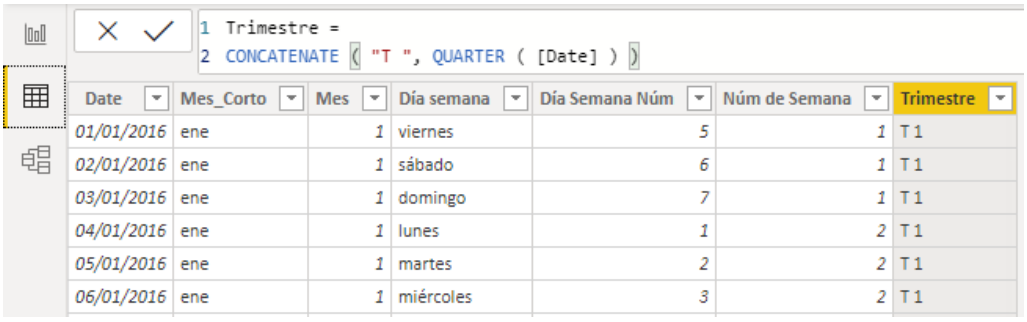
Para que sea más intuitivo, vamos a hacer que no solo se nos muestre el número de trimestre en las celdas, sino que previamente a este, se nos muestre también la letra "T" para entender claramente que nos referimos a trimestres. Para ello, vamos a tener que añadir la función **CONCATENATE** que nos va a permitir unir la letra "T" con el número de trimestre.

Esta función CONCATENATE va a estar formada por dos argumentos, que van a ser los que se unirán en las celdas.

- **Argumento 1:** La letra "T". Va estar en primera posición en la celda.
- **Argumento 2:** La función QUARTER, que nos indica el número de trimestre.

Con lo cual nuestra función completa será la siguiente:

```
Trimestre =
CONCATENATE ( "T ", QUARTER ( [Date] ) )
```



```
1 Trimestre =
2 CONCATENATE ( "T ", QUARTER ( [Date] ) )
```

Date	Mes_Corto	Mes	Día semana	Día Semana Núm	Núm de Semana	Trimestre
01/01/2016	ene	1	viernes	5	1	T 1
02/01/2016	ene	1	sábado	6	1	T 1
03/01/2016	ene	1	domingo	7	1	T 1
04/01/2016	ene	1	lunes	1	2	T 1
05/01/2016	ene	1	martes	2	2	T 1
06/01/2016	ene	1	miércoles	3	2	T 1

9. Añadir una columna para el semestre con ROUNDUP Y DIVIDE

Esta columna nos va a mostrar el **semestre en el que estamos, teniendo en cuenta la fecha de cada una de las filas de la tabla**. La función que vamos a usar es probablemente la más compleja que veremos, ya que requiere un cálculo. La función DAX completa que vamos a usar va a ser la siguiente:

```
Semestre =
CONCATENATE ( ROUNDUP ( DIVIDE ( MONTH ( [Date] ), 6 ), 0 ), " Sem" )
```

Para entenderla, vamos a ver cada una de las funciones y argumentos que la forman:

La **primera función** que vamos a anidar es MONTH, ya que esto nos permite decirle a Power BI que el cálculo que vamos a realizar será en base al número de mes de cada fila. Por lo tanto, un número que va a ir del 1 al 12.

```
Semestre =
CONCATENATE ( ROUNDUP ( DIVIDE ( MONTH ( [Date] ), 6 ), 0 ), " Sem" )
```

La **siguiente función** que vamos a anidar es DIVIDE. Esta la necesitamos para realizar el cálculo. Lo que vamos a hacer específicamente es coger el número de mes (indicado en el primer argumento con la función MONTH) y lo dividiremos entre 6, que es el número que hemos añadido en el argumento.

```
Semestre =
CONCATENATE ( ROUNDUP ( DIVIDE ( MONTH ( [Date] ), 6 ), 0 ), " Sem" )
```

La función DIVIDE está formado por dos argumentos:

- **Argumento 1**: Numerador, formado por MONTH.
- **Argumento 2**: Denominador, introducimos el número 6.

La **tercera función** que vamos a anidar va a ser ROUNDUP. Con esto estaremos indicando que el resultado de la división lo vamos a redondear al alza, de manera que todos los resultados por debajo de 1 se van a redondear a 1 (primer semestre), y todos los resultados por encima de 1 se van a redondear a 2 (segundo semestre).

```
Semestre =
CONCATENATE ( ROUNDUP ( DIVIDE ( MONTH ( [Date] ), 6 ), 0 ), " Sem" )
```

ROUNDUP está formado por dos argumentos:

- **Argumento 1**: Contenido que queremos redondear, en este caso la división calculada.
- **Argumento 2**: Cómo lo queremos redondear, en este caso indicamos con un 0 para no coger decimales. De este modo, el resultado será que el número 1 hará referencia al primer semestre y el número 2 al segundo semestre.

La **última función** que utilizaremos será CONCATENATE, exactamente con la misma finalidad que lo hemos hecho en la columna de trimestre: para unir dos cadenas de texto. En este caso, vamos a añadir la abreviatura “Sem” para visualizar claramente que nos referimos a semestres.

```
Semestre =
CONCATENATE ( ROUNDUP ( DIVIDE ( MONTH ( [Date] ), 6 ), 0 ), " Sem" )
```

CONCATENATE está formada también por dos argumentos:

- **Argumento 1**: El cálculo definitivo hecho y redondeado.
- **Argumento 2**: La abreviatura "Sem" Introducida al final de la función completa.

Una vez realizada la fórmula, se nos crea la nueva columna con el aspecto siguiente:

```
1 Semestre =
2 CONCATENATE ( ROUNDUP ( DIVIDE ( MONTH ( [Date] ), 6 ), 0 ), " Sem" )
```

Date	Mes_Corto	Mes	Día semana	Día Semana Núm	Núm de Semana	Trimestre	Semestre
01/01/2016	ene	1	viernes	5	1	T 1	1 Sem
02/01/2016	ene	1	sábado	6	1	T 1	1 Sem
03/01/2016	ene	1	domingo	7	1	T 1	1 Sem
04/01/2016	ene	1	lunes	1	2	T 1	1 Sem
05/01/2016	ene	1	martes	2	2	T 1	1 Sem
06/01/2016	ene	1	miércoles	3	2	T 1	1 Sem
07/01/2016	ene	1	jueves	4	2	T 1	1 Sem
08/01/2016	ene	1	viernes	5	2	T 1	1 Sem
09/01/2016	ene	1	sábado	6	2	T 1	1 Sem

En este caso vemos que, sin usar una función DAX que nos genere los semestres directamente, hemos podido realizar la operación mediante un cálculo sencillo como es la división y el redondeo.

10. Añadir una columna para el año con YEAR

Esta función es muy sencilla y nos va a permitir tener en una sola **columna el año en el que nos encontramos** en cada una de las filas de la tabla. Para ello vamos a usar la siguiente función DAX:

```
Año =
YEAR ( [Date] )
```

```
1 Año = YEAR([Date])
```

Date	Mes_Corto	Mes	Día semana	Día Semana Núm	Núm de Semana	Trimestre	Semestre	Año
01/01/2016	ene	1	viernes	5	1	T 1	1 Sem	2016
02/01/2016	ene	1	sábado	6	1	T 1	1 Sem	2016
03/01/2016	ene	1	domingo	7	1	T 1	1 Sem	2016
04/01/2016	ene	1	lunes	1	2	T 1	1 Sem	2016
05/01/2016	ene	1	martes	2	2	T 1	1 Sem	2016
06/01/2016	ene	1	miércoles	3	2	T 1	1 Sem	2016
07/01/2016	ene	1	jueves	4	2	T 1	1 Sem	2016
08/01/2016	ene	1	viernes	5	2	T 1	1 Sem	2016
09/01/2016	ene	1	sábado	6	2	T 1	1 Sem	2016

11. Añadir una columna para el día del año con DATEDIFF

Esta nueva columna va a contener **el día del año en el que nos encontramos** en cada una de las filas de la tabla. Para ello, vamos a usar la función **DATEDIFF**. Esta función devuelve el recuento del intervalo entre dos fechas en la unidad de tiempo que indiquemos.

Nosotros buscamos el número de día del año en el que nos encontramos exactamente, para eso hacemos una diferencia entre el día en el que estamos y el primer día del año.

La función DAX completa que vamos a usar es la siguiente:

```
Día del Año =
DATEDIFF ( DATE ( YEAR ( [Date] ), 1, 1 ), [Date], DAY ) + 1
```

Así pues, esta función **DATEDIFF** está formada por tres argumentos:

- **Argumento 1**: La fecha de inicio.
  En nuestro caso, anidamos la función DATE con los siguientes argumentos para obtener el primer día del año:
  1. El primer argumento de DATE es el año: utilizamos la fórmula YEAR para que siempre haga referencia al año correspondiente.
  2. El segundo argumento de DATE es el mes: Ponemos un 1 para indicar enero.
  3. El tercer argumento de DATE es el día: Ponemos un 1 para indicar el primer día del mes.

  ```
  Día del Año =
  DATEDIFF ( DATE ( YEAR ( [Date] ), 1, 1 ), [Date], DAY ) + 1
  ```

- **Argumento 2**: La fecha final. En nuestro caso cogemos la columna "Date" para obtener la fecha de cada una de las filas.

  ```
  Día del Año =
  DATEDIFF ( DATE ( YEAR ( [Date] ), 1, 1 ), [Date], DAY ) + 1
  ```

- **Argumento 3**: Formato del intervalo. En este argumento debemos indicar la unidad de tiempo que queremos que se nos muestre, en nuestro caso DAY (día).

  ```
  Día del Año =
  DATEDIFF ( DATE ( YEAR ( [Date] ), 1, 1 ), [Date], DAY ) + 1
  ```

  Finalmente, **añadimos +1 al final de la fórmula** para sumar un día porque DATEIFF, por defecto, nos calcula la diferencia entre dos fechas.
  Si no añadiéramos el +1, el 1 de enero sería el día 0, ya que la fecha de inicio y fin sería la misma. De esta forma el 1 de enero será el día 1 y el resto de días también aparecerán de forma correcta.

Una vez aplicada la fórmula, se nos genera en nuestra tabla la nueva columna con la función que le hemos añadido:

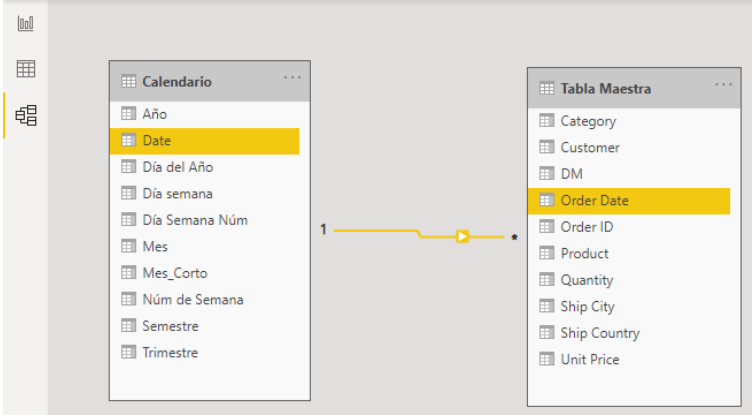
```
1 Día del Año =
2 DATEDIFF ( DATE ( YEAR ( [Date] ), 1, 1 ), [Date], DAY ) + 1
```

Date	Mes_Corto	Mes	Día semana	Día Semana Núm	Núm de Semana	Trimestre	Semestre	Año	Día del Año
25/06/2016	jun	6	sábado	6	26	T 2	1 Sem	2016	177
26/06/2016	jun	6	domingo	7	26	T 2	1 Sem	2016	178
27/06/2016	jun	6	lunes	1	27	T 2	1 Sem	2016	179
28/06/2016	jun	6	martes	2	27	T 2	1 Sem	2016	180
29/06/2016	jun	6	miércoles	3	27	T 2	1 Sem	2016	181
30/06/2016	jun	6	jueves	4	27	T 2	1 Sem	2016	182
01/07/2016	jul	7	viernes	5	27	T 3	2 Sem	2016	183
02/07/2016	jul	7	sábado	6	27	T 3	2 Sem	2016	184
03/07/2016	jul	7	domingo	7	27	T 3	2 Sem	2016	185
04/07/2016	jul	7	lunes	1	28	T 3	2 Sem	2016	186
05/07/2016	jul	7	martes	2	28	T 3	2 Sem	2016	187
06/07/2016	jul	7	miércoles	3	28	T 3	2 Sem	2016	188

RELACIONAR LA TABLA CALENDARIO CON LA TABLA MAESTRA

Con esta columna, ya podríamos decir que tenemos un calendario completo como para empezar a trabajar con nuestros datos. Hemos podido ver como a partir de una simple columna que nos indicaba la fecha, hemos podido ir creando un conjunto de columnas para tener especificados una serie de campos que nos van a servir a posteriori, cuando queramos empezar a trabajar con nuestras visualizaciones.

Ahora lo único que nos faltaría sería **crear una relación** entre la tabla que contiene todo el conjunto de datos y la tabla calendario que hemos creado. Para ello tenemos que acceder a la vista de ***Modelo*** y crear la relación "Date" de la Tabla Calendario hacia "Order Date" de la Tabla Maestra.



Una vez creada esta relación, ya tenemos nuestro modelo preparado para empezar a trabajar mejor con nuestros datos y relacionarlos con un calendario perfectamente estructurado.

REALIZAR TODO CON UNA SOLA FÓRMULA DAX

Otra **forma aún más rápida de crear nuestra tabla calendario** es escribiendo una sola fórmula DAX más compleja.

Veremos cómo podemos reducir todos los pasos anteriores a uno solo **si dominamos el lenguaje DAX.** Aplicamos esta única fórmula en el momento de crear nuestra tabla:

```
Calendario =
ADDCOLUMNS (
   CALENDARAUTO (),
   "Mes_Corto", FORMAT ( [Date], "mmm" ),
   "Mes", MONTH ( [Date] ),
   "Día Semana", FORMAT ( [Date], "dddd" ),
   "Día Semana Núm", WEEKDAY ( [Date], 2 ),
   "Núm de Semana", WEEKNUM ( [Date], 2 ),
   "Trimestre", CONCATENATE ( "T ", QUARTER ( [Date] ) ),
   "Semestre", CONCATENATE ( ROUNDUP ( DIVIDE ( MONTH ( [Date] ), 6 ), 0 ), " Sem" ),
   "Año", YEAR ( [Date] ),
   "Día del Año", DATEDIFF ( DATE ( YEAR ( [Date] ), 1, 1 ), [Date], DAY ) + 1
)
```

Como vemos, con esta fórmula se nos va a generar automáticamente el mismo calendario completo que hemos hecho antes paso a paso.

```
1  Calendario =
2  ADDCOLUMNS (
3   CALENDARAUTO (),
4   "Mes_Corto", FORMAT ( [Date], "mmm" ),
5   "Mes", MONTH ( [Date] ),
6   "Día Semana", FORMAT([Date],"dddd"),
7   "Día Semana Núm", WEEKDAY ( [Date],2 ),
8   "Núm de Semana" , WEEKNUM([Date],2),
9   "Trimestre", CONCATENATE ( "T ", QUARTER ( [Date] ) ),
10  "Semestre",CONCATENATE(ROUNDUP(DIVIDE(MONTH([Date]),6),0)," Sem"),
11  "Año", YEAR([Date]),
12  "Día del Año", DATEDIFF(DATE(YEAR([Date]),1,1),[Date],DAY)+1
13 )
```

Date	Mes_Corto	Mes	Trimestre	Semestre	Año	Día Semana	Día Semana Núm	Núm de Semana	Día del Año
01/07/2016 0:00:00	jul	7	T 3	2 Sem	2016	viernes	5	27	183
02/07/2016 0:00:00	jul	7	T 3	2 Sem	2016	sábado	6	27	184
03/07/2016 0:00:00	jul	7	T 3	2 Sem	2016	domingo	7	27	185

Con este ejemplo, vemos claramente **el tiempo que nos podemos ahorrar si conocemos en más profundidad el lenguaje DAX.** Si tú también quieres dominarlo y empezar a trabajar más rápido**, infórmate sobre nuestro** [Curso práctico de DAX avanzado.]