# Anexo: Sobre la Tabla Calendario

*Calendario, Funciones DAX de fecha y hora. Calendario en lenguaje M*

*Jose Ignacio González Gómez*
*Departamento de Economía Contabilidad y Finanzas - Universidad de La Laguna*
www.jggomez.eu

**v.2.3 – v.2.5**

# INDICE

1 Concepto y generalidades relacionadas con la tabla calendario ..... 2
2 Funciones basicas Calendar() y Calendarauto() ..... 2
3 Resumen, formas de crear una tabla calendario ..... 4
3.1 Usando funciones DAX como CALENDAR ..... 4
3.2 Desde Power Pivot ..... 4
3.3 Código de tabla calendario delimitado entre fechas ..... 5
3.3.1 Codigo 1 ..... 5
3.3.2 Código de tabla calendario desde la fecha indicada hasta hoy ..... 6
4 Calendarios especiales ..... 7
4.1 Calendario de campañas o eventos especiales ..... 7
5 Formatos asociados al calendario ..... 10
5.1 25 formatos de calendario disponible ..... 10
5.2 Formatos de fecha y hora definidos por el usuario ..... 10
6 Problema las fechas (Meses) aparecen desordenadas ..... 12
7 ¿Crear tabla calendario con DAX o con lenguaje M? ..... 13
7.1 Crear la tabla calendario con DAX ..... 13
7.2 Crear la tabla calendario con M (Power Query) ..... 13
7.3 Tabla comparativa Calendario en DAX o en M ..... 13
7.4 ¿Cuál elegir? ..... 14
7.5 Ejemplos de tabla calendario en lenguaje M ..... 14
7.5.1 Ejemplo 1 ..... 14
7.5.2 Ejemplo 2 ..... 15
8 Bibliografía ..... 16

# 1 Concepto y generalidades relacionadas con la tabla calendario

Algunas características, que nos permiten identificarlas

- Contienen una columna con días sin repetir y con un rango de fechas que contenga, la fecha máxima y mínima de la columna de Tipo Fecha a analizar
- Cada año en la Tabla de Calendario, debe contener todos los días del año.

Las tablas de calendario, nos ayudan a analizar columnas de tipo fecha, presentes en tablas transaccionales, por lo que son una muy buena práctica en cualquier modelo de datos.

## 2 Funciones basicas Calendar() y Calendarauto()

En este anexo presentamos diferentes códigos de creacion de tabla calendario que podemos adaptar a nuestros requerimientos.

- CALENDAR(<start_date>, <end_date>)

*Parámetros:*

start_date Cualquier expresión DAX que devuelve un valor datetime.
end_date Cualquier expresión DAX que devuelve un valor datetime.

*Ejemplo*

= CALENDAR (DATE (2015, 1, 1), DATE (2021, 12, 31))

Esta fórmula devuelve una tabla con fechas entre el 1 de enero de 2015 y el 31 de diciembre de 2021

- CALENDARAUTO(<start_date>, <end_date>)

Devuelve una tabla con una sola columna denominada "Date" que contiene un conjunto contiguo de fechas. El rango de fechas se calcula automáticamente según los datos del modelo.

*Ejemplo*

CALENDARAUTO() devuelve todas las fechas entre el 1 de enero de 2010 y el 31 de diciembre de 2011.
CALENDARAUTO(3) devuelve todas las fechas entre el 1 de marzo de 2010 y el 31 de marzo de 2012.

En este ejemplo, MinDate y MaxDate en el modelo de datos son el 1 de julio de 2010 y el 30 de junio de 2011.

Las funciones de fecha y hora ayudan a crear cálculos basados en fechas y horas. Muchas de las funciones son similares a las funciones de fecha y hora de Excel.

| Función | Descripción |
|---|---|
| CALENDAR | Retorna una tabla de una sola columna con fechas las fechas en un inicio y fin. |
| CALENDARAUTO | Retorna una tabla de una sola columna con fechas detectadas automáticamente. |
| DATE | Retorna la fecha dada en tipo fecha y hora. |
| DATEDIFF | Retorna el número de unidades dado un inicio y un final en un intervalo señalado. |
| DATEVALUE | Convierte una fecha en tipo de dato texto a tipo fecha y hora. |

| | |
|---|---|
| DAY | Retorna un número de 1 a 31 que representa el día en el mes. |
| EDATE | Retorna la fecha especificada desfasada un número de meses adelante o atrás. |
| EOMONTH | Retorna la última fecha de un mes dado un desfase de meses adelante o atrás. |
| HOUR | Retorna la hora como un número desde 0 (12:00 a.m) a 23 (11:00 p.m). |
| MINUTE | Retorna un número de 0 a 59 que representa el minuto. |
| MONTH | Retorna un número de 1 a 12 que representa el mes. |
| NOW | Retorna la fecha y hora actual. |
| QUARTER | Retorna un número de 1 a 4 que representa el trimestre en el año. |
| SECOND | Retorna un número de 0 a 59 que representa el segundo. |
| TIME | Toma una hora, minuto y segundo y lo devuelve en tipo de dato fecha y hora. |
| TIMEVALUE | Convierte un texto que representa una hora a forma fecha y hora. |
| TODAY | Retorna la fecha actual. |
| UTCNOW | Retorna la fecha y hora dada en UTC (Tiempo universal coordinado). |
| UTCTODAY | Retorna la fecha dada en UTC (Tiempo universal coordinado). |
| WEEKDAY | Retorna un número que representa el día de la semana de acuerdo al intervalo deseado. |
| WEEKNUM | Realiza un número que representa la semana del año. |
| YEAR | Retorna el año de una fecha. |
| YEARFRAC | Devuelve la fracción del año que representa el número de días completos entre inicio y final. |

# 3 Resumen, formas de crear una tabla calendario

## 3.1 Usando funciones DAX como CALENDAR

Como ya hemos tratado anteriormente la forma mas frecuente es

## 3.2 Desde Power Pivot

Pestaña Power Pivot, grupo Modelo de Datos y Clic en el comando Administrar y en la ventana**, en la ventana de Power Pivot:**

*Pestaña Diseño -> grupo Calendarios – > comando Tabla de Fechas -> elegimos la opción* **Nuevo**

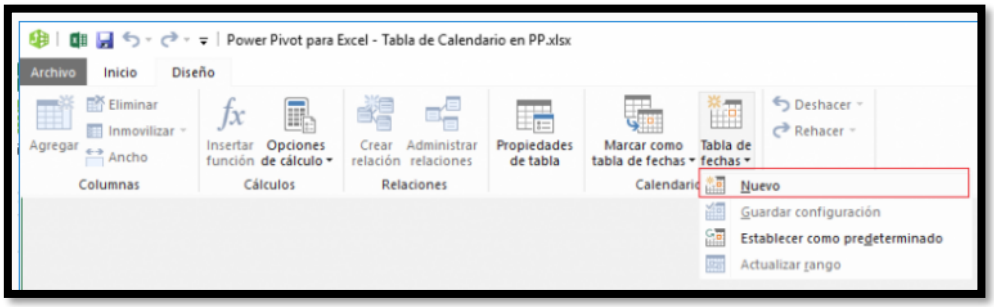



Ilustración 1

Y automáticamente aparece en el Modelo, una tabla de calendario y si vamos a la vista de datos, podemos ver que aparece la tabla, marcada como una **Tabla de Fecha** o **Calendario en Power Pivot.**

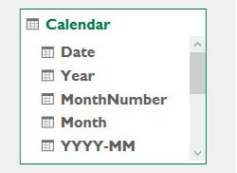



Ilustración 2

**Nota:** *Debemos tener en cuenta que, para crear la tabla de calendario por este método, previamente debe haber cargada al menos una tabla en el modelo, con una columna de Tipo Fecha asignada.*

Por defecto, la tabla se crea con estas columnas, aunque con expresiones DAX, podemos crear otro tipo de columnas calculadas, de ser necesario.

Para editar el rango de la tabla, vamos nuevamente al grupo Calendario, comando Tabla de Fechas, y con la tabla de calendario seleccionada, elegimos la opción Actualizar Rango:

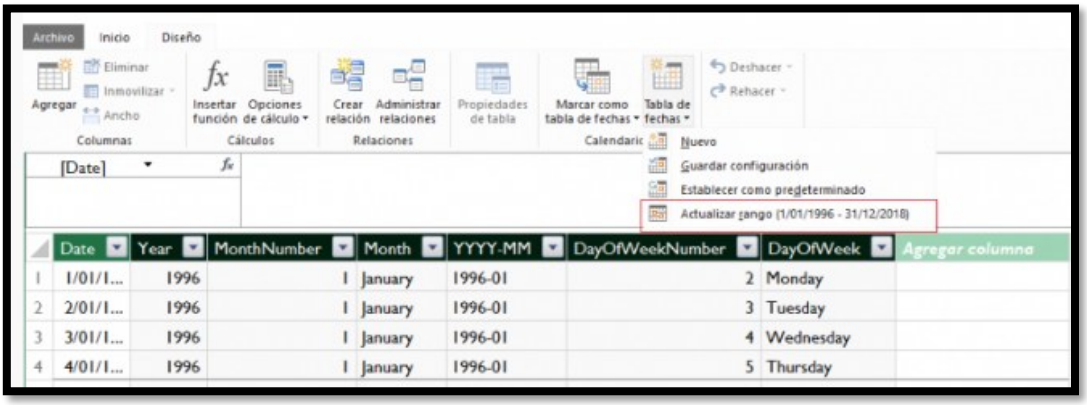



Ilustración 3

## 3.3 Código de tabla calendario delimitado entre fechas

### 3.3.1 Codigo 1

```
dDate =
VAR Fecha = CALENDAR(
    DATE( 2018, 1, 1),
    DATE( 2021, 12, 31)
) return SELECTCOLUMNS(
    Fecha,
    "Fecha", [Date],
    "Año", YEAR( [Date] ),
    "Mes", UPPER( FORMAT( [Date], "MMM" ) ),
    "MesNro", INT( FORMAT( [Date], "M" ) ),
    "NroDia", INT( FORMAT( [Date], "d" ) ),
    "DiaSemana", WEEKDAY( [Date], 2 ),
    "Semana", WEEKNUM( [Date], 2 ),
    "Fin de Semana", WEEKDAY( [Date], 2 ) IN {6,7},
    "Nombre Dia", UPPER( FORMAT( [Date], "DDDD" ) ),
    "TrimNro", ROUNDUP( MONTH( [Date] ) / 3, 0 ),
    "Trimestre", "T" & ROUNDUP( MONTH( [Date] ) / 3, 0 ),
    "SemNro", ROUNDUP( MONTH( [Date] ) / 6, 0),
    "Semestre", "S" & ROUNDUP( MONTH( [Date] ) / 6, 0),
    "Desvio Dia", INT( [Date] ) - INT( Today() ),
    "SortMonthYear", YEAR( [Date] ) * 100 + MONTH( [Date] ),
    "M-Y" , UPPER( FORMAT( [Date], "MMM" ) ) & "-" & RIGHT(YEAR( [Date] ),2),
    "SortTrimYear", YEAR( [Date] ) * 100 + ROUNDUP( MONTH( [Date] ) / 3, 0 ),
    "T-Y" , "T" & ROUNDUP( MONTH( [Date] ) / 3, 0 ) & "-" & RIGHT(YEAR( [Date] ),2),
    "SortSemYear", YEAR( [Date] ) * 100 + ROUNDUP( MONTH( [Date] ) / 6, 0),
    "S-Y" , "S" & ROUNDUP( MONTH( [Date] ) / 6, 0) & "-" & RIGHT(YEAR( [Date] ),2))
```

El resultado obtenido serán las siguientes 2 ilustraciones

| Fecha | Año | Mes | MesNro | NroDia | DiaSemana | Semana | Fin de Semana | Nombre Dia | TrimNro | Trimestre |
|---|---|---|---|---|---|---|---|---|---|---|
| 25/06/18 | 2018 | JUN | 6 | 25 | 1 | 26 | False | LUNES | 2 | T2 |
| 26/06/18 | 2018 | JUN | 6 | 26 | 2 | 26 | False | MARTES | 2 | T2 |
| 27/06/18 | 2018 | JUN | 6 | 27 | 3 | 26 | False | MIÉRCOLES | 2 | T2 |
| 28/06/18 | 2018 | JUN | 6 | 28 | 4 | 26 | False | JUEVES | 2 | T2 |
| 29/06/18 | 2018 | JUN | 6 | 29 | 5 | 26 | False | VIERNES | 2 | T2 |
| 30/06/18 | 2018 | JUN | 6 | 30 | 6 | 26 | True | SÁBADO | 2 | T2 |
| 01/07/18 | 2018 | JUL | 7 | 1 | 7 | 26 | True | DOMINGO | 3 | T3 |
| 02/07/18 | 2018 | JUL | 7 | 2 | 1 | 27 | False | LUNES | 3 | T3 |
| 03/07/18 | 2018 | JUL | 7 | 3 | 2 | 27 | False | MARTES | 3 | T3 |
| 04/07/18 | 2018 | JUL | 7 | 4 | 3 | 27 | False | MIÉRCOLES | 3 | T3 |
| 05/07/18 | 2018 | JUL | 7 | 5 | 4 | 27 | False | JUEVES | 3 | T3 |

| Trimestre | SemNro | Semestre | Desvio Dia | M-Y | T-Y | S-Y | SortMonthYear | SortTrimYear | SortSemYear |
|---|---|---|---|---|---|---|---|---|---|
| T2 | 1 | S1 | -981 | JUN-18 | T2-18 | S1-18 | 201806 | 201802 | 201801 |
| T2 | 1 | S1 | -980 | JUN-18 | T2-18 | S1-18 | 201806 | 201802 | 201801 |
| T2 | 1 | S1 | -979 | JUN-18 | T2-18 | S1-18 | 201806 | 201802 | 201801 |
| T2 | 1 | S1 | -978 | JUN-18 | T2-18 | S1-18 | 201806 | 201802 | 201801 |
| T2 | 1 | S1 | -977 | JUN-18 | T2-18 | S1-18 | 201806 | 201802 | 201801 |
| T2 | 1 | S1 | -976 | JUN-18 | T2-18 | S1-18 | 201806 | 201802 | 201801 |
| T3 | 2 | S2 | -975 | JUL-18 | T3-18 | S2-18 | 201807 | 201803 | 201802 |
| T3 | 2 | S2 | -974 | JUL-18 | T3-18 | S2-18 | 201807 | 201803 | 201802 |
| T3 | 2 | S2 | -973 | JUL-18 | T3-18 | S2-18 | 201807 | 201803 | 201802 |
| T3 | 2 | S2 | -972 | JUL-18 | T3-18 | S2-18 | 201807 | 201803 | 201802 |
| T3 | 2 | S2 | -971 | JUL-18 | T3-18 | S2-18 | 201807 | 201803 | 201802 |

### 3.3.2 Código de tabla calendario desde la fecha indicada hasta hoy

Fuente: clickam.es – Enlace al articulo

```
Calendario=Calendar(MIN(Ventas[Fecha]);Today())
Año=YEAR(Calendario[Fecha])
Mes=MONTH(Calendario[Fecha])
Dia=DAY(Calendario[Fecha])
FechaSk = Calendario[Año]*10000 + Calendario[Mes]*100 + Calendario[Dia]
Semestre=ROUNDUP(Calendario[Mes]/6;0)
Cuatrimestre=ROUNDUP(Calendario[Mes]/4;0)
Trimestre=ROUNDUP(Calendario[Mes]/3;0)
Bimestre=ROUNDUP(Calendario[Mes]/2;0)
Semana = WEEKNUM(Calendario[Fecha])
AñoMes = Calendario[Año] & FORMAT(Calendario[Fecha];"mm")
DiaSemana=WEEKDAY(Calendario[Fecha])
NbAño = Calendario[Año]
NbSemestre = "S"&Calendario[Semestre]
NbSemestreAño = Calendario[NbSemestre] & "/"  & RIGHT(Calendario[NbAño];2)
NbCuatrimestre = "C"&Calendario[Cuatrimestre]
NbCuatrimestreAño = Calendario[NbCuatrimestre] &"/" & RIGHT(Calendario[Año];2)
NbTrimestre = "T"&Calendario[Trimestre]
NbTrimestreAño = Calendario[NbTrimestre] & "/" & RIGHT(Calendario[Año];2)
NbBimestre = "B" & Calendario[Bimestre]
NbBimestreAño = Calendario[NbBimestre] &"/" & RIGHT(Calendario[Año];2)
NbMes = Calendario[Fecha].[Mes]
nbMes3L = left(Calendario[NbMes];3)
NbDia = Calendario[Dia] & " " & Calendario[nbMes3L]
NbSemana = "Sem " & Calendario[Semana] & "/" & RIGHT(Calendario[Año];2)
NdDiaSemana = FORMAT(Calendario[Fecha];"dddd")
```

# 4 Calendarios especiales

## 4.1 Calendario de campañas o eventos especiales

Es corriente que en las empresas nos encontramos con determinadas campañas promocionales como son navidad, Fin de Verano, Black Friday, etc.. que no coinciden exactamente a lo largo de los periodos, ni en fechas ni en duración, pero es importante analizar el resultado de forma comparativa.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Campaña | Año | Fecha Inicio | Fecha Fin |
| 2 | Navidad | 2020 | 05/12/2020 | 30/12/2020 |
| 3 | Navidad | 2021 | 16/12/2021 | 26/12/2021 |
| 4 | Navidad | 2022 | 12/12/2022 | 24/12/2022 |
| 5 | Fin de Verano | 2020 | 01/07/2020 | 18/07/2020 |
| 6 | Fin de Verano | 2021 | 02/07/2021 | 13/07/2021 |
| 7 | Fin de Verano | 2022 | 03/07/2022 | 17/07/2022 |
| 8 | Black Friday | 2020 | 26/11/2020 | 28/11/2020 |
| 9 | Black Friday | 2021 | 24/11/2021 | 24/11/2021 |
| 10 | Black Friday | 2022 | 23/11/2022 | 23/11/2022 |

Campañas | Campañas2 | +

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | Fecha | Campaña | |
| 3 | | 05/12/2020 | Navidad | |
| 4 | | 06/12/2020 | Navidad | |
| 5 | | 07/12/2020 | Navidad | |
| 6 | | 08/12/2020 | Navidad | |
| 7 | | 09/12/2020 | Navidad | |
| 8 | | 10/12/2020 | Navidad | |
| 9 | | 11/12/2020 | Navidad | |
| 10 | | 12/12/2020 | Navidad | |
| 11 | | 13/12/2020 | Navidad | |
| 12 | | 14/12/2020 | Navidad | |

Campañas | Campañas2 | +

**Ilustración 4**

Asi por ejemplo disponemos de las ventas de una empresa a lo largo de varios ejercicios donde disponemos de una serie de periodos de campaña promocional tal y como se muestra en la Ilustración 4 en la hoja "Campañas".

Con el fin de relacionar esta campaña con nuestra tabla calendario proponemos convertir la tabla anterior en una nueva tabla donde disponemos para cada fecha en que dura la campaña la identificación de la misma, tal y como se muestra en la Ilustración 5.

*La única condición que requiere esta tabla que hemos creado es que las fechas deben ser únicas y por tanto no coincidir dos campañas a la vez en fechas.*

Una vez creada esta tabla especial de campañas que podríamos llamarla para nuestro caso "MisCampañasV2" debemos relacionarla a la tabla calendar marcada como tabla de fechas de nuestro modelo de datos de tal forma que nos permita relacionarla con el resto de tablas para calcular la facturación de cada campaña en su respectivo periodo.

Por defecto la relación que se crea es de la tabla de fechas Calendar (1) a la tabla de fechas personalizada "MisCamapañasV2" (muchos) tal y como se muestra en la Ilustración 6.

| | A | B | C |
|---|---|---|---|
| 1 | | | |
| 2 | | Fecha | Campaña |
| 3 | | 05/12/2020 | Navidad |
| 4 | | 06/12/2020 | Navidad |
| 5 | | 07/12/2020 | Navidad |
| 27 | | 29/12/2020 | Navidad |
| 28 | | 30/12/2020 | Navidad |
| 29 | | 16/12/2021 | Navidad |
| 30 | | 17/12/2021 | Navidad |
| 31 | | 18/12/2021 | Navidad |
| 51 | | 23/12/2022 | Navidad |
| 52 | | 24/12/2022 | Navidad |
| 53 | | 01/07/2020 | Fin de Verano |
| 54 | | 02/07/2020 | Fin de Verano |
| 97 | | 17/07/2022 | Fin de Verano |
| 98 | | 26/11/2020 | Black Friday |
| 99 | | 27/11/2020 | Black Friday |
| 100 | | 28/11/2020 | Black Friday |
| 101 | | 24/11/2021 | Black Friday |
| 102 | | 23/11/2022 | Black Friday |

**Ilustración 5**

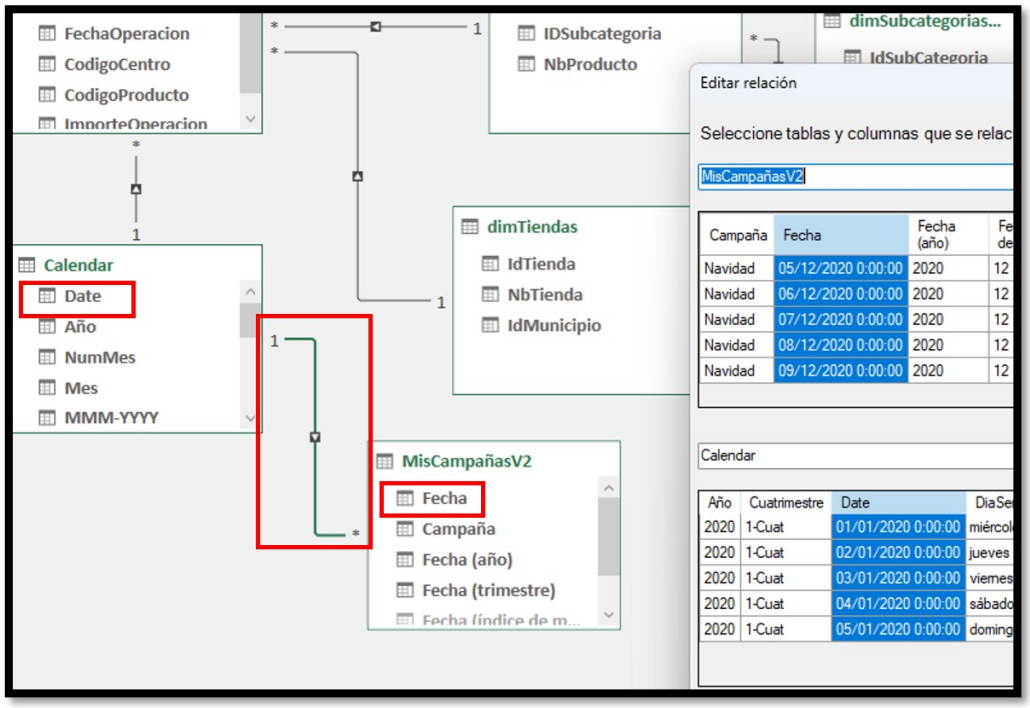



Ilustración 6

Pero en este caso la relación no es correcta… es de 1 a varios, es decir va acoger todas la fechas de calendario y por tanto de factura y las va a sumar dando un error como el que se muestra en la Ilustración 7

Por tanto debemos eliminar la relación anterior y crear una nueva de la tabla MisCampañasV2 a Calendario del tipo 1 a varios, tal y como se propone en la

| Etiquetas de fila | Importe |
|---|---|
| Black Friday | |
| 2020 | 12.017.126 |
| 2021 | 5.621.481 |
| 2022 | 25.099.395 |
| Fin de Verano | |
| 2020 | 12.017.126 |
| 2021 | 5.621.481 |
| 2022 | 25.099.395 |
| Navidad | |
| 2020 | 12.017.126 |
| 2021 | 5.621.481 |
| 2022 | 25.099.395 |
| Total general | 42.738.002 |

Ilustración 7

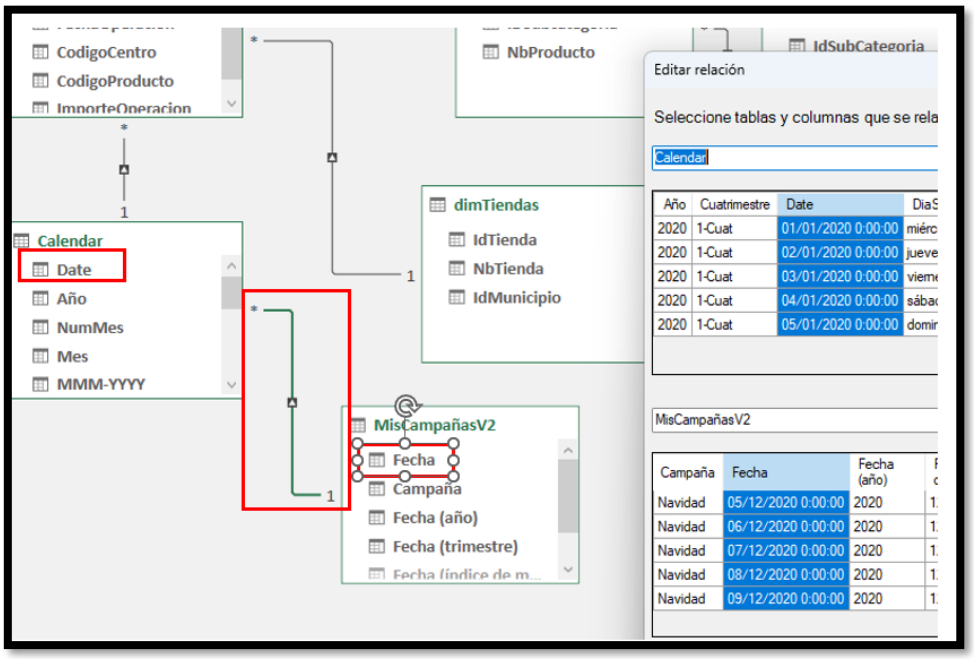



Ilustración 8

Este por tanto es el resultado final deseado

| Campaña | Ingresos | Duración | €/dia |
|---|---|---|---|
| **Black Friday** | | | |
| 2020 | 100.596,23 € | 3 dias | 33.532,08 |
| 2021 | 16.415,14 € | 1 dias | 16.415,14 |
| 2022 | 17.252,96 € | 1 dias | 17.252,96 |
| **Fin de Verano** | | | |
| 2020 | 584.778,37 € | 18 dias | 32.487,69 |
| 2021 | 184.128,03 € | 12 dias | 15.344,00 |
| 2022 | 94.839,16 € | 15 dias | 6.322,61 |
| **Navidad** | | | |
| 2020 | 869.222,82 € | 26 dias | 33.431,65 |
| 2021 | 172.621,29 € | 11 dias | 15.692,84 |
| 2022 | 213.017,51 € | 13 dias | 16.385,96 |
| **Total general** | **2.252.871,51 €** | | |

**NbCategoria**

Atletismo | Baloncesto | Ciclismo
Futbol | Musculación

# 5 Formatos asociados al calendario

## 5.1 25 formatos de calendario disponible

```
1.  Año=YEAR(Calendario[Fecha])
2.  Mes=MONTH(Calendario[Fecha])
3.  Dia=DAY(Calendario[Fecha])
4.  FechaSk = Calendario[Año]*10000 + Calendario[Mes]*100 + Calendario[Dia]
5.  Semestre=ROUNDUP(Calendario[Mes]/6;0)
6.  Cuatrimestre=ROUNDUP(Calendario[Mes]/4;0)
7.  Trimestre=ROUNDUP(Calendario[Mes]/3;0)
8.  Bimestre=ROUNDUP(Calendario[Mes]/2;0)
9.  Semana = WEEKNUM(Calendario[Fecha])
10. AñoMes = Calendario[Año] & FORMAT(Calendario[Fecha];"mm")
11. DiaSemana=WEEKDAY(Calendario[Fecha])
12. NbAño = Calendario[Año]
13. NbSemestre = "S"&Calendario[Semestre]
14. NbSemestreAño = Calendario[NbSemestre] & "/"  & RIGHT(Calendario[NbAño];2)
15. NbCuatrimestre = "C"&Calendario[Cuatrimestre]
16. NbCuatrimestreAño = Calendario[NbCuatrimestre] &"/" & RIGHT(Calendario[Año];2)
17. NbTrimestre = "T"&Calendario[Trimestre]
18. NbTrimestreAño = Calendario[NbTrimestre] & "/" & RIGHT(Calendario[Año];2)
19. NbBimestre = "B" & Calendario[Bimestre]
20. NbBimestreAño = Calendario[NbBimestre] &"/" & RIGHT(Calendario[Año];2)
21. NbMes = Calendario[Fecha].[Mes]
22. nbMes3L = left(Calendario[NbMes];3)
23. NbDia = Calendario[Dia] & " " & Calendario[nbMes3L]
24. NbSemana = "Sem " & Calendario[Semana] & "/" & RIGHT(Calendario[Año];2)
25. NdDiaSemana = FORMAT(Calendario[Fecha];"dddd")
```

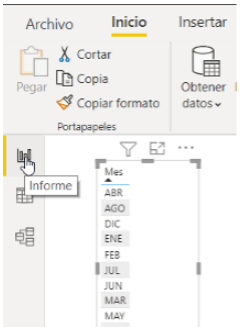
## 5.2 Formatos de fecha y hora definidos por el usuario

| | |
|---|---|
| c | Muestra la fecha como ddddd y muestra la hora como ttttt, en ese orden. Muestra solo la información de fecha si no hay ninguna parte fraccionaria en el número de serie de fecha; muestra solo la información de hora si no hay ninguna parte entera. |
| d | Muestra el día como un número sin un cero a la izquierda (1–31). |
| dd | Muestra el día como un número con un cero a la izquierda (01–31). |
| ddd | Muestra el día abreviado (dom–sáb). Localizado. |
| dddd | Muestra el día como un nombre completo (domingo–sábado). Localizado. |
| ddddd | Muestra la fecha como una fecha completa (día, mes y año), con el formato que especifique la configuración de formato de fecha corta del sistema. El formato de fecha corta predeterminado es m/d/yy. |
| dddddd | Muestra un número de serie de fecha como fecha completa (día, mes y año), con el formato de fecha larga que reconozca el sistema. El formato de fecha larga predeterminado es mmmm dd, yyyy. |
| w | Muestra el día de la semana como un número (del 1 para el domingo al 7 para el sábado). |
| ww | Muestra la semana del año como un número (1–54). |
| m | Muestra el mes como un número sin un cero a la izquierda (1–12). Si m sigue inmediatamente a h o hh, se muestra el minuto en lugar del mes. |
| mm | Muestra el mes como un número con un cero a la izquierda (01–12). Si m sigue inmediatamente a h o hh, se muestra el minuto en lugar del mes. |
| mmm | Muestra el mes abreviado (ene–dic). Localizado. |
| mmmm | Muestra el mes con el nombre completo (enero–diciembre). Localizado. |
| q | Muestra el trimestre del año como un número (1–4). |
| y | Muestra el día del año como un número (1–366). |

| | |
|---|---|
| yy | Muestra el año como un número de 2 dígitos (00–99). |
| yyyy | Muestra el año como un número de 4 dígitos (100–9999). |
| h | Muestra la hora como un número sin un cero a la izquierda (0–23). |
| hh | Muestra la hora como un número con un cero a la izquierda (00–23). |
| n | Muestra el minuto como un número sin un cero a la izquierda (0–59). |
| nn | Muestra el minuto como un número con un cero a la izquierda (00–59). |
| d | Muestra el segundo como un número sin un cero a la izquierda (0–59). |
| ss | Muestra el segundo como un número con un cero a la izquierda (00–59). |
| ttttt | Muestra la hora como una hora completa (hora, minuto y segundo), utilizando el separador de hora definido en el formato de hora que reconozca el sistema. Se muestra un cero a la izquierda si se selecciona la opción de cero a la izquierda y la hora es anterior a las 10:00. o p. m. El formato predeterminado es h:mm:ss. |
| AM/PM | Usa el reloj de 12 horas y muestra AM en mayúsculas con cualquier hora antes del mediodía; muestra PM en mayúsculas con cualquier hora entre el mediodía y las 11:59 p.m. |
| am/pm | Usa el reloj de 12 horas y muestra AM en minúsculas con cualquier hora antes del mediodía; muestra PM en minúsculas con cualquier hora entre el mediodía y las 11:59 p. m. |
| A/P | Usa el reloj de 12 horas y muestra una A mayúscula con cualquier hora antes del mediodía; muestra una P mayúscula con cualquier hora entre el mediodía y las 11:59 p. m. |
| a/p | Usa el reloj de 12 horas y muestra una A minúscula con cualquier hora antes del mediodía; muestra una P minúscula con cualquier hora entre el mediodía y las 11:59 p. m. |
| AMPM | Usa el reloj de 12 horas y muestra el literal de cadena AM tal y como lo define el sistema con cualquier hora antes del mediodía; muestra el literal de cadena PM tal y como lo define el sistema con cualquier hora entre el mediodía y las 11:59 p. m. AMPM puede estar en mayúsculas o en minúsculas, pero coincidirá con la cadena tal y como la defina la configuración del sistema. El formato predeterminado es AM/PM. Si el sistema se define en el reloj de 24 horas, la cadena se establece normalmente en una cadena de longitud cero. |

# 6 Problema las fechas (Meses) aparecen desordenadas

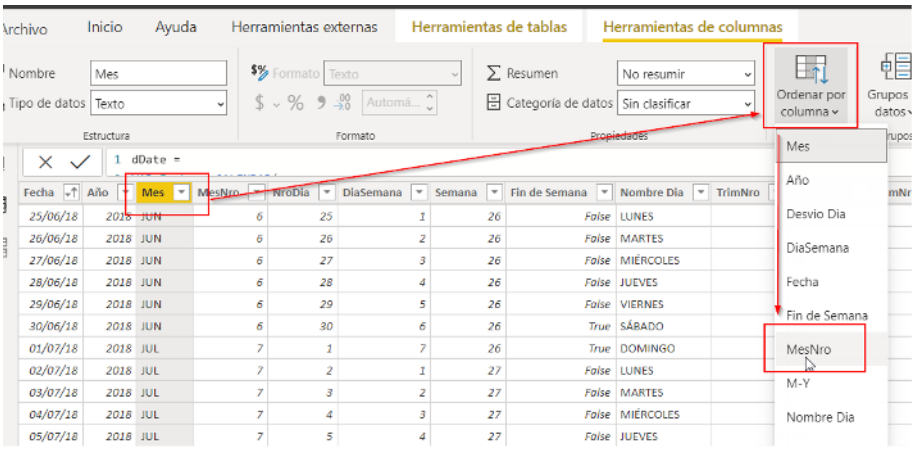
Fuente: Tabla Calendario con DAX en PowerBI y funciones de Inteligencia de tiempo - Olivia



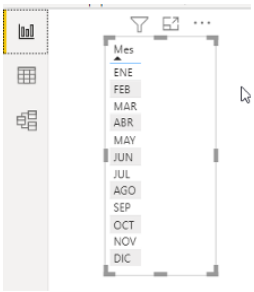

Puede ser que en algunas ocasiones sobre todo cuando trabajamos con un elemento visual como una tabla las fechas aparezcan desordenadas, es decir por orden alfabético y no por el orden fecha

La solución será ordenar aquellas columnas que se ordenan en función de otra columna, por ejemplo la columna "Mes". ¿Por qué es necesaria la ordenación de la columna "Mes" en función de otra columna? Porque por defecto, el sistema nos la ordena de forma alfabética. Veamos qué pasa si arrastramos la variable "Mes" en el lienzo de PowerBI y lo mostramos en una tabla. Para ir al lienzo, hemos ido al botón de "Informe", justo encima del de "Datos".

Para ello lo que haremos será en la opción de "Datos" donde se visualizan las tablas, seleccionaremos la columna de "Mes" y cuando la cabecera de la columna se quede resaltada de color amarillo, pulsaremos la opción "Ordenar por columna" y seleccionaremos la opción "MesNro". De esta forma la columna de "Mes" nos quedará ordenada por número de mes y no por orden alfabético.




Si volvemos al informe, veremos cómo los meses ya nos aparecen ordenados según el orden natural de tiempo establecido. Igualmente podemos hacer lo propio para las columnas como:




- Nombre de dia – Ordenado por Dia Semana
- Trimestre – Ordenado por TrimNro
- Semestre – Ordenado por SemNro.. etc

# 7 ¿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:

## 7.1 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.

## 7.2 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.

## 7.3 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). |

## 7.4 ¿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.

## 7.5 Ejemplos de tabla calendario en lenguaje M

### 7.5.1 Ejemplo 1

```
let
   Source = #date(2018, 1, 1),
   Custom1 = List.Dates(Source, Number.From(#date(2023, 1, 1))- Number.From(Source)
,#duration(1,0,0,0)),
   #"Converted to Table" = Table.FromList(Custom1, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
   #"Renamed Columns" = Table.RenameColumns(#"Converted to Table",{{"Column1", "Date"}}),
   #"Changed Type" = Table.TransformColumnTypes(#"Renamed Columns",{{"Date", type date}}),
   #"Inserted Year" = Table.AddColumn(#"Changed Type", "Year", each Date.Year([Date]), type
number),
   #"Inserted Quarter" = Table.AddColumn(#"Inserted Year", "Quarter", each
Date.QuarterOfYear([Date]), type number),
   #"Quarter Name" = Table.AddColumn(#"Inserted Quarter", "QuarterName", each "Q" &
Number.ToText([Quarter])),
   #"Inserted Month" = Table.AddColumn(#"Quarter Name", "Month", each Date.Month([Date]),
type number),
   #"Inserted Month Name" = Table.AddColumn(#"Inserted Month", "MonthName", each
Date.MonthName([Date]), type text),
   #"Month Short" = Table.AddColumn(#"Inserted Month Name", "MonthShort", each
Date.ToText([Date],"MMM")),
   MonthYear = Table.AddColumn(#"Month Short", "MonthYear", each Date.ToText([Date],"MMM
yy")),
```

```
  #"Month Sort" = Table.AddColumn(MonthYear, "MonthYearSort", each Date.ToText([Date],
"yyyyMM")),
  #"Inserted Day Name" = Table.AddColumn(#"Month Sort", "DayName", each
Date.DayOfWeekName([Date]), type text),
  #"Day Name Short" = Table.AddColumn(#"Inserted Day Name", "DayNameShort", each
Date.ToText([Date],"ddd")),
  #"Inserted Day of Week" = Table.AddColumn(#"Day Name Short", "DayOfWeek", each
Date.DayOfWeek([Date],Day.Saturday), type number),
  #"Inserted Week of Year" = Table.AddColumn(#"Inserted Day of Week", "WeekOfYear", each
Date.WeekOfYear([Date]), type number),
  #"Inserted Start of Week" = Table.AddColumn(#"Inserted Week of Year", "StartOfWeek", each
Date.StartOfWeek([Date],Day.Saturday), type date),
  #"Inserted End of Week" = Table.AddColumn(#"Inserted Start of Week", "EndOfWeek", each
Date.EndOfWeek([Date],Day.Saturday), type date)
in
  #"Inserted End of Week"
```

### 7.5.2 Ejemplo 2

```
let
  FechaInicio = #date(2018, 1, 1),
  FechaFin = #date(Date.Year(DateTime.LocalNow()) + 1, 12, 31),
  Dias = List.Dates(FechaInicio, Duration.Days(FechaFin - FechaInicio) + 1, #duration(1, 0,
0, 0)),
  TablaFechas = Table.FromList(Dias, Splitter.SplitByNothing(), {"Fecha"}),
  TipoFecha = Table.TransformColumnTypes(TablaFechas, {{"Fecha", type date}}),
  ConColumnas = Table.AddColumn(TipoFecha, "Año", each Date.Year([Fecha])),
  Final = Table.AddColumn(ConColumnas, "Mes", each Date.Month([Fecha]))
in
```

### 7.5.3 Ejemplo 3

(1) Cómo crear una Tabla Calendario con Lenguaje M en Power Query - YouTube

```
let
  FechaInicio = #date(2015, 1, 1),
  FechaFin = #date(2030, 12, 31),
  Dias = Duration.Days(FechaFin - FechaInicio),
  Fechas = List.Dates(FechaInicio, Dias + 1, #duration(1,0,0,0)),
  Tabla = Table.FromList(Fechas, Splitter.SplitByNothing(), {"Fecha"}),

  ConColumnas = Table.TransformColumns(Tabla,{{"Fecha", each _, type date}}),
  AgregarColumnas = Table.AddColumn(ConColumnas, "Año", each Date.Year([Fecha]),
Int64.Type),
  ConMes = Table.AddColumn(AgregarColumnas, "Mes", each Date.Month([Fecha]),
Int64.Type),
  ConNombreMes = Table.AddColumn(ConMes, "NombreMes", each Date.ToText([Fecha],
"MMMM"), type text),
  ConTrimestre = Table.AddColumn(ConNombreMes, "Trimestre", each
Date.QuarterOfYear([Fecha]), Int64.Type),
  ConDia = Table.AddColumn(ConTrimestre, "Día", each Date.Day([Fecha]), Int64.Type),
  ConNombreDia = Table.AddColumn(ConDia, "NombreDía", each Date.ToText([Fecha],
"dddd"), type text),
  ConDiaSemana = Table.AddColumn(ConNombreDia, "DíaSemana", each
Date.DayOfWeek([Fecha])+1, Int64.Type),
  ConSemanaAno = Table.AddColumn(ConDiaSemana, "SemanaAño", each
Date.WeekOfYear([Fecha]), Int64.Type),
  ConDiaAno = Table.AddColumn(ConSemanaAno, "DíaAño", each
Date.DayOfYear([Fecha]), Int64.Type),
```

```
ConFinSemana = Table.AddColumn(ConDiaAno, "EsFinDeSemana", each if
Date.DayOfWeek([Fecha]) >= 5 then true else false, type logical)
in
```

# 8 Bibliografía

CALENDARAUTO | Interactive Chaos
Formatos personalizados de fecha y hora
Personalizar Formatos de Fecha y Hora Con La Función FORMAT (DAX)
Jggomez "sobre la tabla calendario"
**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
(1) Cómo crear una Tabla Calendario con Lenguaje M en Power Query - YouTube