



# DAX (III)

## Calendario, Funciones de Inteligencia de Tiempo

Jose Ignacio González Gómez

Departamento de Economía Contabilidad y Finanzas - Universidad de La Laguna

[www.jggomez.eu](http://www.jggomez.eu)

1. La tabla calendario como base de las funciones de inteligencia de tiempo
  - a. Introducción. Importancia de contar con una tabla calendario
  - b. Creación de la tabla calendario con DAX, caso Power BI
  - c. Marcar la tabla como tabla de fechas y Ordenar las columnas del calendario
2. Sobre las funciones de inteligencia de tiempo
  - a. Creando funciones de inteligencia de tiempo con medidas rápidas.
  - b. SAMEPERIODLASTYEAR. Compara importes en el mismo periodo
  - c. TOTALYTD, TOTALQTD y TOTALMTD. Para importes acumulados
  - d. DATESYTD – DATESQTD- DATESMTD, evolucionadas a TOTALYTD – TOTALQTD - TOTALMTD
  - e. DATEDIFF, para intervalos de fechas cálculo de la media de días.
  - f. DATEADD, para comparar con dos o tres años antes. Calcular tasa de crecimiento
  - g. DATESBETWEEN
  - h. Otras funciones
3. Anexos
  - a. Código de tabla calendario delimitado entre fechas
  - b. Código de tabla calendario desde la fecha indicada hasta hoy
  - c. 25 formatos de calendario disponible
  - d. Formatos de fecha y hora definidos por el usuario

# 1. La tabla calendario como base de las funciones de inteligencia de tiempo

## 1.1. Introducción

Es habitual en nuestros informes la necesidad de trabajar con datos temporales para el análisis entre periodos como años, meses, semanas, trimestres, día de la semana, etc..

Tenemos a disposición en DAX un conjunto de funciones de inteligencia de tiempo y que, entre otras cosas nos van a permitir:

- Calcular acumulados y realizar comparaciones entre periodos
- Conocer la variación de las ventas de este año respecto el año anterior
- Calcular la variación en % respecto la semana pasada de la medida a analizar
- Nos permite identificar los días de fin de semana o entre semana.
- El disponer de una tabla calendario nos permitirá vincular simultáneamente distintas tablas, de tal forma que el filtro de fecha puede ser común en más de una tabla facilitando la navegabilidad en el informe.

Lo expuesto justifica la relevancia de contar en nuestro modelo de datos con la tabla calendario y en concreto cuando la variable tiempo (fecha) sea relevante en nuestros análisis al permitir trabajar con las funciones de inteligencia de tiempo.

Pero debemos tener presente que la tabla de calendario tiene sentido si está relacionada con por lo menos una tabla transaccional o de hechos (también llamada base o tabla matriz o maestra)

El Lenguaje DAX no es ajeno a ello, por eso trae un paquete especial de funciones que permite reportes **que de otra manera serían muy difícil de lograr, en ocasiones hasta imposibles. Esto es Time Intelligence**, una categoría de funciones DAX especializadas en tratar con las fechas.

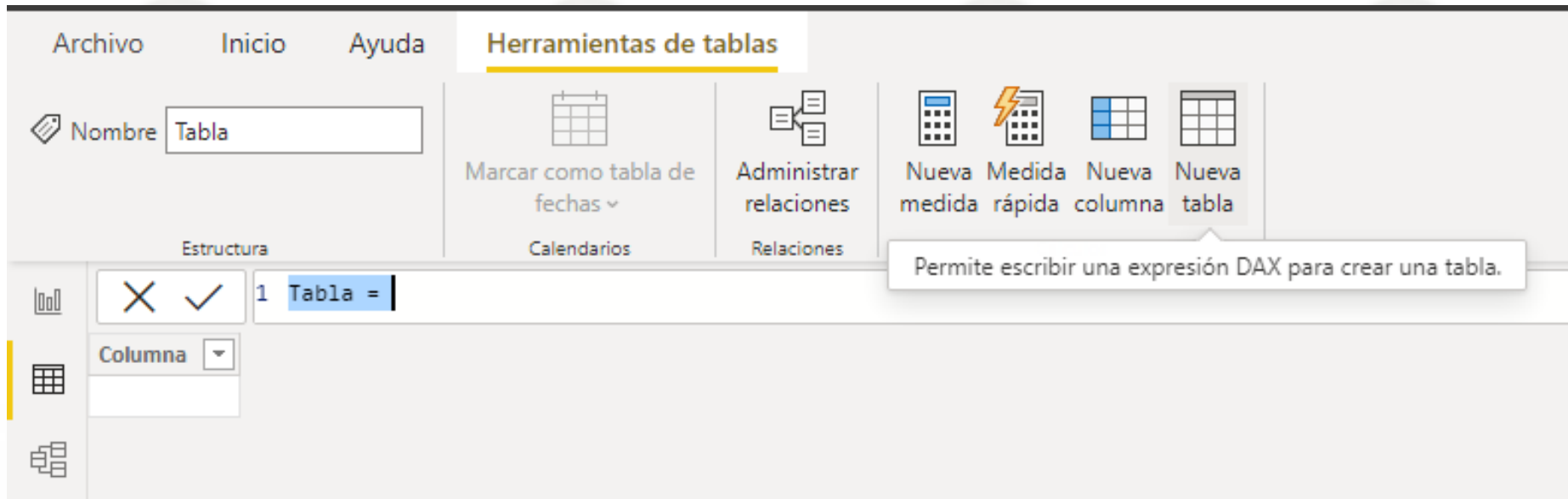
## 1.2. Creacion de la tabla calendario con DAX, caso Power BI

La creacion de la tabla calendario la podemos realizar por varias vías:

- En Power Query con la opcion Fechas y tendremos a disposición las columnas de fechas necesaria pudiendo ser ampliada y personalizadas con día de semana, día del año, trimestre, etc.
- En Power Pivot creando tabla calendario a traves de un acceso directo
- Empleando fórmulas DAX en Power BI

### Creacion de la tabla calendario con DAX, caso Power BI

En el caso de Power BI, para crear la tabla calendarios accedemos al apartado “DATOS” “Herramientas Tablas” y una vez seleccionada, marcaremos la opción “Nueva Tabla”, y aquí escribiremos el código que nos va a permitir generar una tabla calendario personalizada.



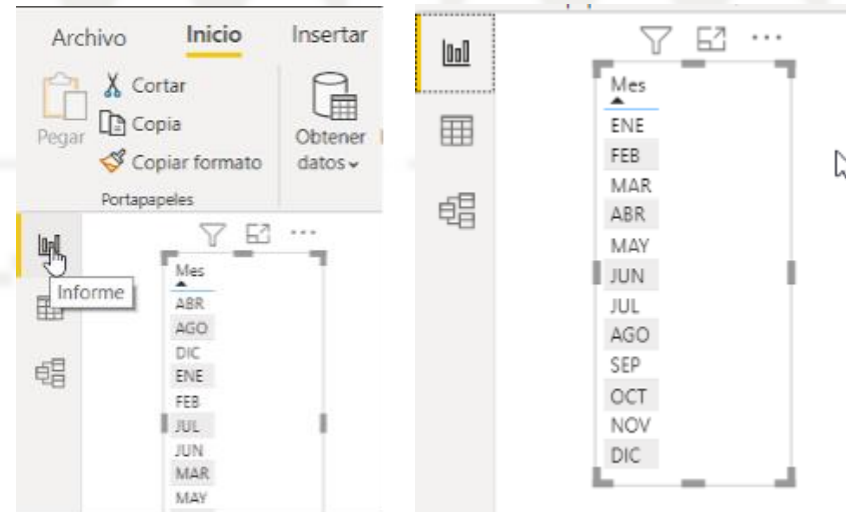
Hay dos funciones DAX para crear la tabla calendario pero CALENDAR() es la más usada y en el Anexo de este documento presentamos diferentes códigos de creación de tabla calendario que podemos adaptar a nuestras necesidades.

Otra alternativa es el uso de la función CALENDARAUTO() que detecta los campos tipo Fecha de otras tablas para crear la tabla de fechas. Esta opción presenta como ventaja que se adapta a las fechas de nuestra tabla de hecho.

De todas formas, existen otras formas de acceso a la creación de tablas con DAX que conlleva el mismo procedimiento de creación de código que explicaremos a continuación.

Una vez **creada la tabla necesitamos realizar una serie de ajustes que pasamos a comentar.**

- **Marcar la tabla como tabla de fechas.** Para ello tenemos varias vías por ejemplo cuando estemos visualizando los datos de la tabla, marcaremos la opción de la pestaña de la cinta de opciones “Herramientas de tablas” y pulsaremos la opción “Marcar como tabla de fechas” – “Marcar como tabla de fechas”.
- **Ordenar las columnas del calendario.** El segundo ajuste será ordenar aquellas columnas que se ordenan en función de otra columna, por ejemplo, la columna “Mes”. Porque por defecto, el sistema la ordena alfabética.



**VER ANEXO – CODIGO o BUSCAR EN INTERNET**

- **Relacionar la tabla calendario.** Finalmente debemos incorporar esta tabla de fechas al modelo de datos y relacionarla con un campo fecha normalmente de la tabla de hechos

## 2. Sobre las funciones de inteligencia de tiempo

### 2.1 Introducción. Creando funciones de inteligencia de tiempo con medidas rápidas

Las funciones de Inteligencia de Tiempo (Time Intelligence) son un grupo especial de funciones que no tiene ningún equivalente con las funciones de fecha y hora en Excel, deben ser consideradas como funciones de fecha y hora de “élite”.

Este tipo de funciones nos permitirá dar respuesta a cuestiones como las siguientes:

- ¿Cómo está funcionando el negocio en relación con el mismo período del año pasado?
- ¿Cuáles han sido nuestras Ventas de año hasta la fecha (YTD) al 1 de junio?
- ¿Cuál fue el mejor trimestre en los últimos dos años?

Destacar que ***no es necesario aprender todos los detalles de todas estas funciones, solo es necesario saber que existen para poder aplicarlas en el momento oportuno.***

Presentamos a continuación las más frecuentes.

***Estas funciones se combinan con otras como son CALCULATE ()***

Tenemos la posibilidad de crear medidas de inteligencia de tiempo **con la opción Medidas Rápidas** que a través de un asistente nos permitirá generar la medida deseada y para ello accedemos desde el menú Inicio - Calculo seleccionamos medidas rápidas que nos da acceso a un asistente para generarla, lo primero es seleccionar el calculo a realizar del grupo interesado en nuestro caso grupo de inteligencia de tiempo – Cambio de año a año, y lo que se solicita es:

1. Valor base o medida que se va a utilizar para nuestro caso será cantidad de la tabla ventas. Es decir valor del que queremos calcular el cambio
2. Fecha, en este caso como valor de fecha seleccionamos la columna fecha de la tabla calendario
3. Numero de periodos pasados en los que queremos calcular el cambio.

## 2.2 SAMEPERIODLASTYEAR. Compara importes en el mismo periodo

***Posiblemente sea la función de inteligencia de tiempo más utilizada.*** Gracias a este tipo de uso, *podemos comparar un importe entre un periodo de tiempo*, con su importe en el mismo periodo de tiempo, pero del año anterior.

Para utilizarla simplemente tenemos que **crear una medida**, y en la parte del filtro poner la función **SAMEPERIODLASTYEAR** con el parámetro de tipo fecha por el que queramos comparar  
*Sintaxis:* SAMEPERIODLASTYEAR (dates)

### **Ejemplo I: Ventas diarias del año pasado** ([Enlace](#))

Creamos una medida para comparar las ventas actuales con las del año pasado. Aplicaremos SAMEPERIODLASTYEAR combinando la función CALCULATE que recordemos nos permite cambiar el contexto de cualquier cálculo.

Por último, ***necesitamos hacer referencia a la columna Fechas de la tabla Tiempo.***

Ventas del Año Pasado =

```
CALCULATE([Venta Total],SAMEPERIODLASTYEAR(Tiempo[Fecha]))
```

Fecha	Venta Total	Ventas del Año Pasado
domingo, 1 de enero de 2017	\$280,650.09	\$306,288.84
lunes, 2 de enero de 2017	\$563,182.46	\$667,560.05
martes, 3 de enero de 2017	\$321,628.96	\$800,599.29
miércoles, 4 de enero de 2017	\$80,320.91	\$205,994.60
jueves, 5 de enero de 2017	\$682,575.17	\$106,780.43
viernes, 6 de enero de 2017	\$985,562.55	\$451,197.13
sábado, 7 de enero de 2017	\$676,083.75	\$862,581.55

## Ejemplo II: Ventas mensuales año anterior ([Enlace](#))

Disponemos de una tabla con ventas y definimos la medida Ventas (como total ventas) de la siguiente forma:

**Ventas = SUM(FactSales[SalesAmount])**

A continuación, creamos una medida para calcular las ventas del mismo período que el implicado en el contexto un año antes:

**Ventas año anterior = CALCULATE([Ventas], SAMEPERIODLASTYEAR(DimDate[Datekey]))**

Creamos una tabla con estas dos medidas y vemos el resultado 

Year	Month	Ventas	Ventas año anterior
2007	January	193.305.554,64 €	
	February	209.439.067,93 €	
	March	203.991.979,69 €	
	April	276.891.048,16 €	
	May	288.749.508,61 €	
	June	283.186.644,54 €	
	July	272.818.635,11 €	
	August	263.780.279,28 €	
	September	257.282.781,95 €	
	October	288.853.903,92 €	
	November	308.752.784,65 €	
	December	297.341.103,65 €	
	<b>Total</b>		<b>3.144.393.292,13 €</b>
2008	January	183.970.020,28 €	193.305.554,64 €
	February	191.106.948,30 €	209.439.067,93 €
	March	183.393.312,89 €	203.991.979,69 €
	April	223.849.292,33 €	276.891.048,16 €
	May	220.502.302,24 €	288.749.508,61 €
	June	214.455.381,54 €	283.186.644,54 €
	July	246.239.251,91 €	272.818.635,11 €
	August	231.189.642,07 €	263.780.279,28 €
	September	227.942.617,84 €	257.282.781,95 €
	October	211.203.579,42 €	288.853.903,92 €
	November	247.706.608,64 €	308.752.784,65 €
	December	260.854.259,59 €	297.341.103,65 €
	<b>Total</b>		<b>2.642.413.217,03 €</b>
2009	January	183.941.322,57 €	183.970.020,28 €
	February	180.954.406,26 €	191.106.948,30 €
	March	180.981.062,98 €	183.393.312,89 €
	April	211.894.484,93 €	223.849.292,33 €
	May	234.303.813,59 €	220.502.302,24 €
	June	227.822.106,40 €	214.455.381,54 €

## 2.3 TOTALYTD, TOTALQTD y TOTALMTD. Para importes acumulados

Cuando tratamos de importes acumulados debemos pensar siempre en las siguientes funciones:

*TOTALYTD devuelve una tabla que contiene las fechas desde el principio de año del contexto actual hasta la última fecha del contexto actual.*

*TOTALQTD igualmente devuelve una tabla que contiene las fechas desde el principio del trimestre del contexto actual hasta la última fecha del del contexto actual.*

***TOTALMTD** será igual que las anteriores una tabla única con las fechas desde el principio de mes del contexto actual hasta la última fecha del contexto actual.*

Este trio de funciones retorna una tabla con las fechas desde el principio de año, trimestre o mes hasta el día de “hoy”” respectivamente (Según el título) y *permite acumular por año, trimestre o mes.*

Estas tres funciones (TOTALYTD, TOTALMTD y TOTALQTD) **son una evolución de las tres funciones** (DATESYTD – DATESQTD- DATESMTD).

### **Concepto y sintaxis.**

Vamos a explicar la sintaxis para el caso de TOTALQTD, para el resto es equivalente solo cambiando el contexto.

*Sintaxis:*

TOTALQTD(expression,dates,[ filter])

*Argumentos de la función:*

- **expression:** Expresión que devuelve un escalar.
- **dates:** Columna conteniendo fechas.
- **filter:** Argumento opcional. Filtro a aplicar al contexto actual.



## Ejemplo I: TOTALYTD ([Enlace](#))

Disponemos de una tabla con ventas y definimos la medida ventas totales como:

$$\text{Ventas} = \text{SUM}(\text{Ventas}[\text{Importe}])$$

A continuación, definimos la medida Ventas YTD que calcula el total de ventas desde principio de año del contexto actual hasta la última fecha del contexto actual:

$$\text{Ventas YTD} = \text{TOTALYTD}(\text{Ventas}[\text{Ventas}], \text{Calendario}[\text{Fecha}])$$

Si llevamos esta última medida a una matriz, obtenemos:

Fecha	Importe
<i>martes, 15 de septiembre de 2015</i>	2
<i>miércoles, 30 de septiembre de 2015</i>	3
<i>lunes, 5 de octubre de 2015</i>	1
<i>miércoles, 7 de octubre de 2015</i>	2
<i>miércoles, 11 de noviembre de 2015</i>	3
<i>viernes, 1 de enero de 2016</i>	5
<i>viernes, 15 de enero de 2016</i>	2
<i>miércoles, 3 de febrero de 2016</i>	5
<i>miércoles, 17 de febrero de 2016</i>	1
<i>viernes, 1 de abril de 2016</i>	2
<i>sábado, 21 de mayo de 2016</i>	3
<i>viernes, 15 de julio de 2016</i>	4
<i>miércoles, 28 de septiembre de 2016</i>	2



Mes Año	Ventas YTD
2015-09	5
2015-10	8
2015-11	11
2015-12	11
2016-01	7
2016-02	13
2016-03	13
2016-04	15
2016-05	18
2016-06	18
2016-07	22
2016-08	22
2016-09	24
2016-10	24
2016-11	24

Puede observarse cómo la medida se reinicializa a cero al principio de cada año

## **Ejemplo II: TOTALQTD** ([Enlace](#))

La medida "Ventas QTD" calcula el total de ventas desde principio del trimestre del contexto actual hasta la última fecha del contexto actual:

Disponemos de una tabla con ventas y definimos la medida ventas totales como:

$$\text{Ventas QTD} = \text{TOTALQTD}(\text{Ventas}[\text{Ventas}], \text{Calendario}[\text{Fecha}])$$

Se observa cómo la medida se reinicializa a cero al principio de cada trimestre.

Mes Año	Ventas QTD
2015-09	5
2015-10	3
2015-11	6
2015-12	6
2016-01	7
2016-02	13
2016-03	13
2016-04	2
2016-05	5
2016-06	5
2016-07	4
2016-08	4
2016-09	6

## **Ejemplo III: TOTALMTD** ([Enlace](#))

La medida "Ventas MTD" calcula el total de ventas desde principio del mes del contexto actual hasta la última fecha del contexto actual.

$$\text{Ventas MTD} = \text{TOTALMTD}(\text{Ventas}[\text{Ventas}], \text{Calendario}[\text{Fecha}])$$

Puede observarse cómo la medida se reinicializa a cero al principio de cada mes

Fecha	Ventas MTD
viernes, 15 de enero de 2016	7
sábado, 16 de enero de 2016	7
domingo, 17 de enero de 2016	7
miércoles, 3 de febrero de 2016	5
jueves, 4 de febrero de 2016	5
viernes, 5 de febrero de 2016	5
sábado, 6 de febrero de 2016	5
viernes, 1 de abril de 2016	2
sábado, 2 de abril de 2016	2
domingo, 3 de abril de 2016	2
lunes, 4 de abril de 2016	2
martes, 5 de abril de 2016	2
miércoles, 6 de abril de 2016	2

## 2.4 DATEDIFF, para intervalos de fechas cálculo de la media de días.

A través de esta función podemos averiguar el intervalo de tiempo que hay entre dos fechas. Los intervalos de tiempo que nos puede devolver esta función son los siguientes:

• SECOND	• HOUR	• WEEK	• QUARTER
• MINUTE	• DAY	• MONTH	• YEAR

Uno de los casos prácticos donde se utiliza DATEDIFF es para el cálculo de la media por un intervalo de tiempo específico, ya que, por defecto, la tabla de hechos está relacionada con la dimensión de tiempo por una columna de tipo Date por lo que nos calcularía la media por día.

### **Concepto y sintaxis.**

DATEDIFF (start\_date,end\_date,interval)

### ***Argumentos de la función:***

- start\_date: Fecha de comienzo en formato datetime.
- end\_date: Fecha final en formato datetime.
- interval: Intervalo a usar para el cálculo. Puede tomar cualquiera de los valores expresados anteriormente (Second, Hour, Minuete, Day, Week,...)

Es importante destacar que la función devuelve la diferencia entre las dos fechas (el número de intervalos de diferencia), no el número de intervalos cubiertos (que puede ser mayor). Así, la diferencia entre el 31 de diciembre de 2015 y el 1 de enero de 2017 no son tres años (aun cuando la primera fecha es de 2015 y la última 2017), sino dos años (que es la diferencia entre 2017 y 2015)

## **Ejemplo I: DATEDIFF** ([Enlace](#))

El siguiente ejemplo compara la fecha del 31 de diciembre de 2015 a las 23:59:59 con el 1 de enero de 2016 a las 0:0:0 (un segundo después). Como puede verse, la diferencia entre ambas fechas según todos los intervalos (salvo para la semana) es 1

Diferencia = `DATEDIFF(Tabla1[Fecha de inicio];Tabla1[Fecha final];SECOND)`

Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	1

Diferencia = `DATEDIFF(Tabla1[Fecha de inicio];Tabla1[Fecha final];MINUTE)`

Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	1

Diferencia = `DATEDIFF(Tabla1[Fecha de inicio];Tabla1[Fecha final];HOUR)`

Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	1

Diferencia = `DATEDIFF(Tabla1[Fecha de inicio];Tabla1[Fecha final];DAY)`

Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	1

Diferencia = `DATEDIFF`(Tabla1[Fecha de inicio];Tabla1[Fecha final];`WEEK`)

Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	0

Diferencia = `DATEDIFF`(Tabla1[Fecha de inicio];Tabla1[Fecha final];`MONTH`)

Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	1

Diferencia = `DATEDIFF`(Tabla1[Fecha de inicio];Tabla1[Fecha final];`QUARTER`)

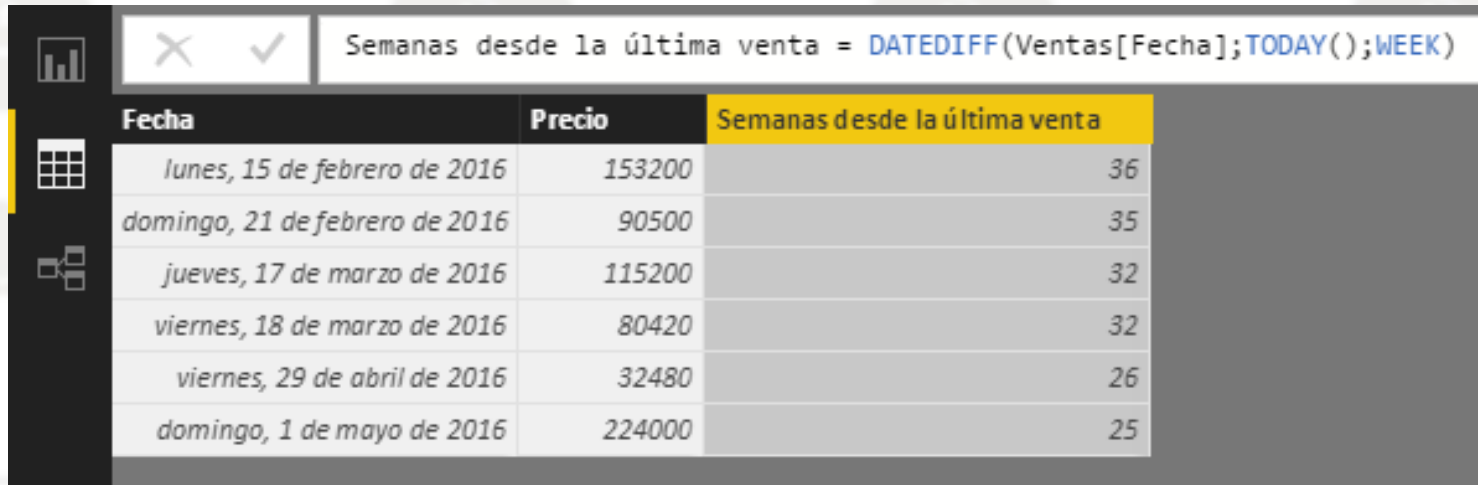
Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	1

Diferencia = `DATEDIFF`(Tabla1[Fecha de inicio];Tabla1[Fecha final];`YEAR`)

Fecha de inicio	Fecha final	Diferencia
31/12/2015 23:59:59	01/01/2016 0:00:00	1

## **Ejemplo II: DATEDIFF** Semana desde la última venta ([Enlace](#))

En este ejemplo se muestra el número de semanas transcurridas desde la fecha de venta hasta el día de hoy (cálculo que implica las funciones DATEDIFF y TODAY):



The screenshot shows a Power BI table with three columns: 'Fecha', 'Precio', and 'Semanas desde la última venta'. The formula bar at the top displays the DAX formula: `Semanas desde la última venta = DATEDIFF(Ventas[Fecha];TODAY();WEEK)`. The table contains six rows of sales data, with the calculated column values ranging from 25 to 36 weeks.

Fecha	Precio	Semanas desde la última venta
<i>lunes, 15 de febrero de 2016</i>	153200	36
<i>domingo, 21 de febrero de 2016</i>	90500	35
<i>jueves, 17 de marzo de 2016</i>	115200	32
<i>viernes, 18 de marzo de 2016</i>	80420	32
<i>viernes, 29 de abril de 2016</i>	32480	26
<i>domingo, 1 de mayo de 2016</i>	224000	25

## 2.5 DATEADD: % de tasa de crecimiento

Esta función nos **va a permitir** seleccionar el marco de tiempo de retraso que deseamos, es decir cuánto tiempo atrás quiero moverme en términos de plazos, por ejemplo tres días, meses, trimestres o años.

### Concepto y sintaxis.

DATEADD(dates, number\_of\_intervals, Interval)

### *Argumentos de la función:*

- dates: Columna conteniendo fechas.
- number\_of\_intervals: Número de intervalos a desplazar hacia adelante o hacia atrás en el tiempo el período implicado en el contexto actual. Si es positivo el valor será hacia delante en el tiempo y si es negativo hacia atrás.
- interval: Tipo de intervalo. Los valores posibles son: DAY, MONTH, QUARTER o YEAR, y deberán escribirse en mayúsculas y sin comillas

Conociendo esta función podemos crear la tasa de crecimiento básica, la cual compare el año actual con el año previo, a diferencia de la tasa de crecimiento básica que creamos anteriormente año que toma un año base.

**Ejemplo I: DATEADD** comparar las ventas actuales con las de un periodo anterior concreto (por ejemplo dos años, tres, etc..) ([Enlace](#))

En este ejemplo vamos a determinar las ventas para el mes anterior usando DATEADD:

```
Ventas Anteriores =  
    CALCULATE([Venta Total],DATEADD(Tiempo[Fecha],-1,MONTH))
```

Una vez que añado esta medida a una tabla, podemos comenzar a comparar los ingresos totales con los ingresos del mes pasado.

Fecha	Venta Total	Ventas Anteriores
lunes, 6 de febrero de 2017	\$580,649.73	\$985,562.5475
martes, 7 de febrero de 2017	\$305,270.59	\$676,083.745
miércoles, 8 de febrero de 2017	\$60,339.67	\$727,026.24
jueves, 9 de febrero de 2017	\$960,407.92	\$725,093.875
domingo, 5 de marzo de 2017	\$652,914.62	\$786,390.97
lunes, 6 de marzo de 2017	\$375,968.82	\$580,649.7275
martes, 7 de marzo de 2017	\$301,815.58	\$305,270.59
<b>Total</b>	<b>\$808,639,703.54</b>	<b>\$804,429,375.23</b>



**Ejemplo II: DATEADD** Venta del mes vs Venta de mes anterior. Crecimiento respecto a un periodo anterior ([Enlace](#))

Definimos la medida Ventas (como total ventas) de la siguiente forma:

**Ventas = SUM(FactSales[SalesAmount])**

Creamos a continuación la medida ventas de mes anterior

**Ventas mes anterior = CALCULATE([Ventas], DATEADD(DimDate[Datekey], -1, MONTH))**

Si llevamos ambas medidas a la cabecera de columnas en una matriz con los meses en filas, el resultado es el siguiente:

CalendarMonth ▲	Ventas	Ventas mes anterior
200701	193.305.554,64 €	
200702	209.439.067,93 €	193.305.554,64 €
200703	203.991.979,69 €	209.439.067,93 €
200704	276.891.048,16 €	203.991.979,69 €
200705	288.749.508,61 €	276.891.048,16 €
200706	283.186.644,54 €	288.749.508,61 €

**Ejemplo III: DATEADD** Ejemplo, crecimiento respecto a un periodo ([Enlace](#))

Un escenario frecuente es el que supone el cálculo de crecimiento de una métrica expresado como porcentaje (ventas, número de clientes, número de unidades vendidas, etc.) con respecto a un cierto período anterior, por ejemplo un año o un trimestre, por poner un par de ejemplos. El cálculo básico a realizar es el siguiente:

**[% crecimiento] = (Valor actual - Valor hace un año) / Valor hace un año**

## 2.6 DATESBETWEEN

Esta función devuelve una tabla conteniendo una única columna de fechas que comienza y termina con las fechas incluidas como parámetros, es decir, permite un control total sobre el intervalo de fechas de una medida.

### **Concepto y sintaxis.**

DATESBETWEEN( dates, start\_date, end\_date)

#### *Argumentos de la función:*

- **dates:** Referencia a una columna que contenga fechas
- **start\_date:** Fecha inicial a considerar.
- **end\_date:** Fecha final a considerar.

### **Ejemplo I: DATESBETWEEN** Ventas entre dos periodos concretos ([Enlace](#))

Definimos la siguiente medida en la que calculamos las ventas entre el 2-01-2007 y el 4-01-2017:

Ventas período = CALCULATE( [Ventas], DATESBETWEEN( FactSales[DateKey],DATE(2007,1,2), DATE(2007,1,4)))

Si mostramos una tabla con las ventas por día y una segunda visualización de tipo "tarjeta" con la medida Ventas período recién calculada:

DateKey	SalesAmount
lunes, 1 de enero de 2007	6.085.839,18 €
<b>martes, 2 de enero de 2007</b>	<b>6.270.657,17 €</b>
<b>miércoles, 3 de enero de 2007</b>	<b>6.096.024,11 €</b>
<b>jueves, 4 de enero de 2007</b>	<b>5.979.164,13 €</b>
viernes, 5 de enero de 2007	5.926.584,02 €
sábado, 6 de enero de 2007	6.150.610,75 €
domingo, 7 de enero de 2007	6.517.040,34 €
lunes, 8 de enero de 2007	5.856.724,52 €
martes, 9 de enero de 2007	6.184.820,44 €
miércoles, 10 de enero de 2007	6.612.222,96 €
jueves, 11 de enero de 2007	6.524.046,81 €
viernes, 12 de enero de 2007	5.889.201,15 €
sábado, 13 de enero de 2007	6.330.346,71 €
<b>Total</b>	<b>8.341.224.364,83 €</b>

**18.345.845,41 €**  
Ventas período

Podemos ver como los dos últimos argumentos indican expresiones que retornen una fecha. Si queremos colocar una fecha fija, por ejemplo, 01/04/2000, entonces lo debemos hacer mediante una de las funciones clásicas de fecha y hora: DATE, así: DATE (2000;4;1). Aunque en los argumentos se puede indicar como: "1/4/2000", evita esta forma ya que puede retornar valores inesperados e incorrectos.

## 2.7 Otras funciones

Otras funciones que también nos pueden servir de ayuda para realizar consultas DAX son:

- TODAY: Devuelve la fecha actual.
- NOW: Devuelve la fecha actual con horas, minutos y segundos.
- FIRSDATE: Devuelve la primera fecha.
- LASTDATE: Devuelve la ultima fecha.
- FIRSTNONBLANK: Devuelve el primer valor que no está en blanco.
- LASTNONBLANK: Devuelve el ultimo valor que no está en blanco.
- ENDOFMONTH, ENDOFQUARTER & ENDOFYEAR: Devuelve el final del mes, cuatrimestre o año.
- STARTOFMONTH, STARTOFQUARTER & STARTOFYEAR: Devuelve el inicio del mes, trimestre o año.
- PREVIOUSDAY, PREVIOUSMONTH, PREVIOUSQUARTER & PREVIOUSYEAR - NEXTDAY, NEXTMONTH, NEXTQUARTER & NEXTYEAR . Estas funciones, como bien indica el nombre, devuelven la información del día, mes, cuatrimestre o año anterior.
- NEXTDAY, NEXTMONTH, NEXTQUARTER & NEXTYEAR son sus equivalentes para el siguiente día, mes, cuatrimestre o año.

### 3 Anexo

En este anexo presentamos diferentes códigos de creación de tabla calendario que podemos adaptar a nuestros requerimientos. **Para copiar el código acceder al documento completo [enlace](#):**

*Código de tabla calendario delimitado entre fechas – resultado siguientes 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

*Código de tabla calendario desde la fecha indicada hasta hoy*

*25 formatos de calendario disponible*

*Formatos de fecha y hora definidos por el usuario*

**Todo ello disponible en el documento [enlace](#):**

## **Referencias**

<https://aleson-itc.com/%F0%9F%95%92-trabajando-con-funciones-de-time-intelligence-serie-dimdate-2-3/>

Construcción de tablas de calendario y porqué son la bomba

<https://www.excelfreeblog.com/tablas-de-calendario-porque-son-la-bomba/>

Funciones DAX de inteligencia de tiempo

[https://interactivechaos.com/es/recursos-educativos/funciones-dax?title=&field\\_funcion\\_dax\\_categoria\\_value%5B%5D=Funciones+de+inteligencia+de+tiempo](https://interactivechaos.com/es/recursos-educativos/funciones-dax?title=&field_funcion_dax_categoria_value%5B%5D=Funciones+de+inteligencia+de+tiempo)

Inteligencia de tiempo en Power Pivot en Excel

<https://support.office.com/es-es/article/inteligencia-de-tiempo-en-power-pivot-en-excel-016acf7b-9ded-411e-ba6c-ed8b8c368011?ui=es-ES&rs=es-ES&ad=ES>