



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

INDICE

1	La tabla calendario como base de las funciones de inteligencia de tiempo.....	3
1.1	Introducción.....	3
1.2	La tala calendario, aspectos generales.....	3
1.2.1	Importancia de contar con una tabla calendario.....	3
1.2.2	Tipos de calendario. “Calendario Estándar” versus “Calendario Personalizado”.....	4
1.3	Creacion de la tabla calendario.....	5
1.3.1	Introducción.....	5
1.3.2	Creacion de la tabla calendario con DAX, caso Power BI.....	5
1.3.3	Marcar la tabla como tabla de fechas.....	5
1.3.4	Ordenar las columnas del calendario.....	6
1.4	Incorporación al modelo de datos.....	7
1.4.1	Relacionar la tabla calendario.....	7
1.4.2	Múltiples tablas de calendario en el mismo modelo.....	8
2	Sobre las funciones de inteligencia de tiempo.....	9
2.1	Presentación.....	9
2.2	Creando funciones de inteligencia de tiempo con medidas rápidas.....	9
2.3	SAMEPERIODLASTYEAR. Compara importes en el mismo periodo.....	10
2.3.1	Concepto y sintaxis.....	10
2.3.2	Ejemplo: Ventas diarias del año pasado.....	10
2.3.3	Ejemplo: Ventas mensuales año anterior.....	11
2.4	TOTALYTD, TOTALQTD y TOTALMTD. Para importes acumulados.....	12
2.4.1	Concepto y sintaxis.....	12
2.4.2	Ejemplo TOTALYTD.....	12
2.4.3	Ejemplos TOTALQTD.....	13
2.4.4	Ejemplos TOTALMTD.....	13
2.4.5	Ejemplo: DATESYTD junto a TOTALYTD.....	14

2.5	<i>DATESYTD – DATESQTD- DATESMTD, evolucionadas a TOTALYTD – TOTALQTD - TOTALMTD</i>	14
2.5.1	Concepto y sintaxis de estas funciones.....	14
2.5.2	Ejemplo DATESYTD	14
2.5.3	Ejemplo DATESQTD.....	15
2.5.4	Ejemplo DATESMTD.....	15
2.6	<i>DATEDIFF, para intervalos de fechas cálculo de la media de días.</i>	16
2.6.1	Concepto y sintaxis.....	16
2.6.2	Ejemplo I.....	16
2.6.3	Ejemplo Semana desde la última venta.	17
2.7	<i>DATEADD, por ejemplo para comparar con dos o tres años antes. Para calcular tasa de crecimiento.</i>	18
2.7.1	Concepto y sintaxis.....	18
2.7.2	Ejemplo, comparar las ventas actuales con las de un periodo anterior concreto (por ejemplo dos años, tres, etc.).....	18
2.7.3	Ejemplo básicos I . Venta del mes vs Venta de mes anterior. Crecimiento respecto a un periodo anterior.....	19
2.7.4	Ejemplo, crecimiento respecto a un periodo anterior.....	19
2.8	<i>DATESBETWEEN</i>	19
2.8.1	Concepto y sintaxis.....	19
2.8.2	Ejemplo Base	20
2.8.3	Ejemplo1: Tabla Dinámica con Medida Vida de Operación.....	20
2.8.4	Ejemplo 2.....	21
2.9	<i>Otras funciones</i>	21
3	Anexos.....	22
1.5	<i>Código de tabla calendario delimitado entre fechas</i>	22
1.6	<i>Código de tabla calendario desde la fecha indicada hasta hoy</i>	23
1.7	<i>25 formatos de calendario disponible</i>	23
1.8	<i>Formatos de fecha y hora definidos por el usuario</i>	24
4	Referencias	25

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 encaminadas a moldear fechas y permitir 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.

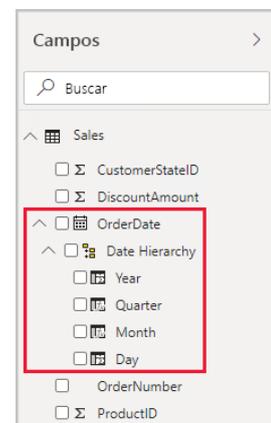
1.2 La tala calendario, aspectos generales

1.2.1 Importancia de contar con una tabla calendario

Gran parte de los análisis de datos implica explorar y comparar con relación a fechas y horas. Tanto si va a sumar importes de ventas del último año fiscal o a comparar el crecimiento intermensual, es fundamental que incluya fechas en los informes. Uno de los mayores atractivos de Power Pivot es que le permite hacer que el modelo de datos sea realmente eficaz a la hora de trabajar con fechas y horas.

Es cierto que la funcionalidad de **Inteligencia de Tiempo** (Auto Date-Time en inglés) permite que cualquier Campo Fecha se convierta en una jerarquía de tiempo o “mini tabla de tiempo oculta” y podríamos pensar que con esta opción activa (se activa por defecto) ya tendríamos la jerarquía de fechas solucionada y no necesitaríamos una tabla específica de fechas, por tanto nos podríamos plantear ¿Es necesario una tabla de fechas independiente?.

Definitivamente SÍ., y entre razones especialmente porque al usar fórmulas de tiempo en DAX es recomendable tener una tabla de fechas independiente para su correcto funcionamiento.



En casos como SAMEPERIODLASTYEAR o similares si hacemos el filtro de tiempo directamente en la tabla de hechos seguramente no devolverá nada, ya que no puede mirar el año anterior si está filtrada por el año actual.

1.2.2 Tipos de calendario. “Calendario Estándar” versus “Calendario Personalizado”

Calendarios estándar, es sobre el que básicamente centraremos nuestra atención y explicaciones, y que se caracteriza por las siguientes propiedades:

- Febrero tiene 28 días (29 en años bisiestos) en el mismo, y todos los otros meses tienen 30 o 31 días en ellos
- Trimestres constan de tres meses consecutivos - meses cuyas longitudes se han descrito anteriormente
- Años tienen 365 días en ellos (366 en años bisiestos)
- Un mes dado este año podría tener más / menos sábados (o cualquier otro día) en él que en el mismo mes del año anterior

En otras palabras, un calendario estándar es el calendario que tienes colgado en la pared.

Las funciones de inteligencia de tiempo de Power Pivot operan bajo el supuesto de que se utiliza un calendario estándar.

Pero debemos considerar algunas limitaciones relacionadas con el calendario estándar y que condicionan nuestros análisis y conclusiones como son:

- Comparando este mes con el mes pasado a menudo no es “justo” cuando el mes pasado tuvo 31 días y éste tiene 30, por ejemplo. ¿Realmente realizamos 3% peor este mes o es que sólo se debe a la diferente cantidad de días?
- Incluso dos meses de la misma longitud a menudo no son comparaciones justas ya que contienen diferentes números de días de fin de semana frente a días de la semana.

En determinados sector y contexto es necesario contar con un **Calendario Personalizado**, como, por ejemplo:

- A veces, la unidad de tiempo medido ni siquiera se parece al calendario de pared – “Semestres” en el mundo académico y “Temporadas” en el mundo del deporte, por ejemplo.
- Yendo más lejos, a veces (como en la ciencia), queremos comparar literalmente períodos de tiempo en lugar de períodos de calendario – tales como “los primeros cinco minutos después de un evento” en comparación con los siguientes quince minutos etc.
- Los calendarios personalizados son frecuentes también, por ejemplo:
 - Temporadas Deportivas
 - Programación de Proveedores
 - Semestres Académicos
 - Calendarios de Minoristas.

En esta línea señalar la necesidad de un calendario personalizado en el ámbito, es el calendario personalizado 4-4-5, que es frecuente para algunas industrias de manufactura. El calendario 4-4-5 divide un año en 4 trimestres, donde cada uno está compuesto por 13 semanas las cuales se agrupan en dos periodos de 4 semanas y uno de 5 semanas, estos hacen las veces de los meses.

Las tablas de calendarios personalizados pueden estar configuradas como nosotros necesitamos, es decir, puede que las columnas de fechas no tengan fechas consecutivas, que las fechas correspondan a periodos o ciclos asignados en vez de meses. Por ejemplo, un calendario universitario, las primeras 5 semanas a partir de febrero pueden corresponder a la categoría primer tercio, luego 4 semanas a segundo tercio y otras 5 semanas posteriores a tercer tercio.

1.3 Creacion de la tabla calendario

1.3.1 Introducción

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

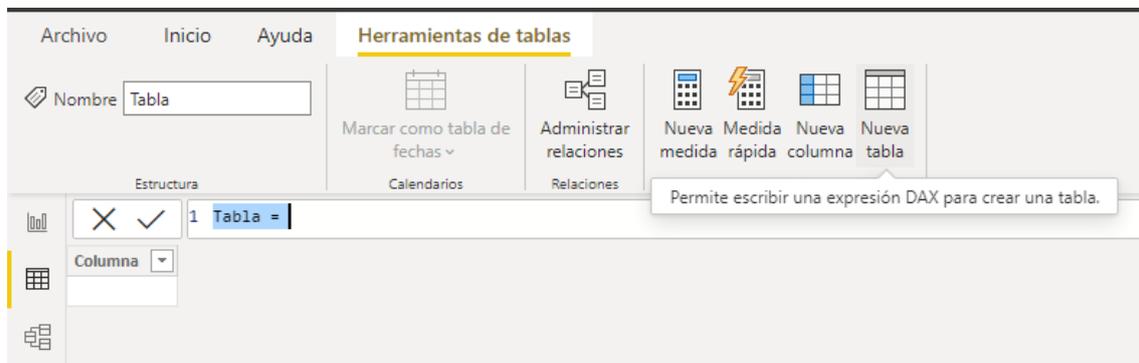
- En Power Query con la opcion Fechas y tendremos a disposición las columnas de fechas necesaria pudiendo ser ampliada y personalizadas con día de semana, día del año, trimestre, etc.
- En Power Pivot creando tabla calendario a traves de un acceso directo
- Empleando fórmulas DAX en Power BI

Destacar que en cualquiera de los casos anteriores podemos acceder a la opcion DAX para crear la tabla Calendario

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

Fuente: [Olivia.es](#) - [Enlace al articulo](#)

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 creacion 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 opcion presenta como ventaja que se adapta a las fechas de nuestra tabla de hecho.

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

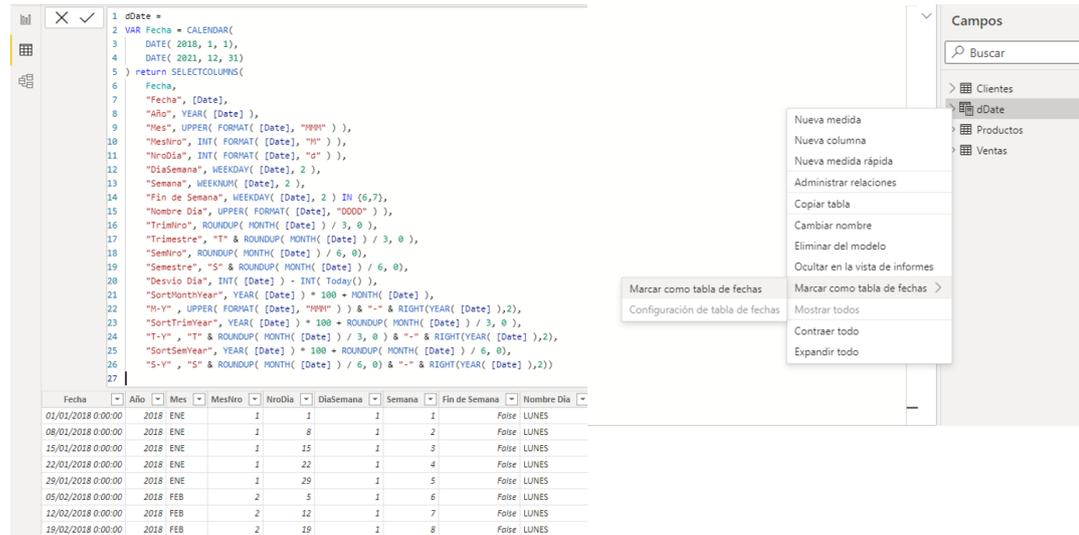
Una vez creada la tabla necesitamos realizar una serie de ajustes que pasamos a comentar.

1.3.3 Marcar la tabla como tabla de fechas.

El primer paso es establecer esta tabla como tabla de fechas y para hacerlo 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”.

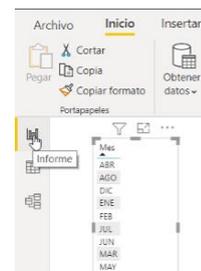


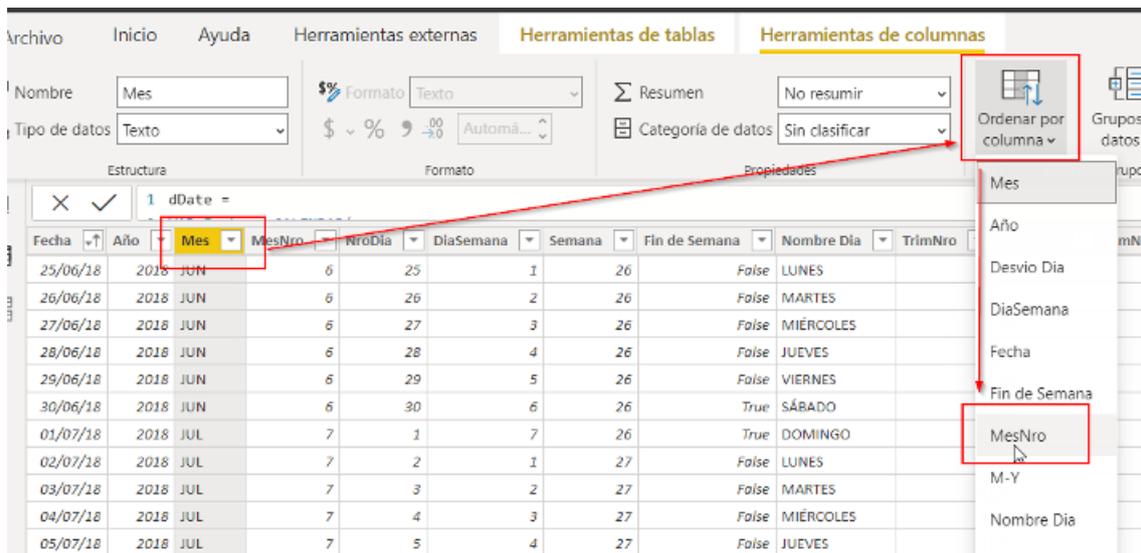
O bien simplemente seleccionando la tabla en el modelo de datos y elegir la opción correspondiente tal y como se muestra en la siguiente ilustración.



1.3.4 Ordenar las columnas del calendario

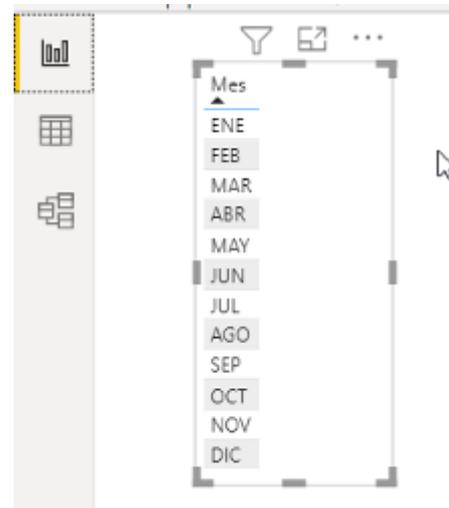
El segundo ajuste importante será ordenar aquellas columnas que se ordenan en función de otra columna, por ejemplo, la columna “Mes”. ¿Por qué es necesaria la ordenación de la columna “Mes” en función de otra columna? Porque por defecto, el sistema nos la ordena de forma alfabética. Veamos qué pasa si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





Evidentemente, tener ordenados los datos de esta forma no nos está dando ninguna información útil, para ello lo que haremos será volver a 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.



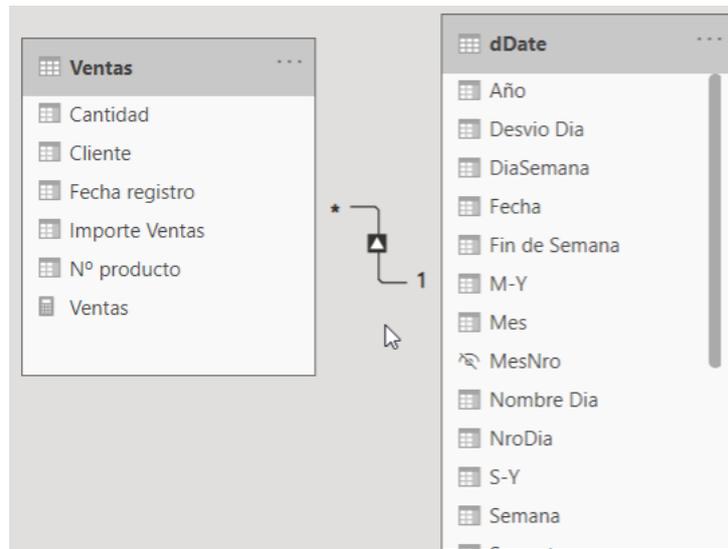
A continuación, vamos a hacer lo propio para las siguientes columnas:

- Nombre de día – Ordenado por Día Semana
- Trimestre – Ordenado por TrimNro
- Semestre – Ordenado por SemNro
- M-Y – Ordenado por SortMonthYear
- T-Y – Ordenado por SortTrimYear
- S-Y – Ordenado por SortSemYear

1.4 Incorporación al modelo de datos

1.4.1 Relacionar la tabla calendario

Finalmente debemos incorporar esta tabla de fechas al modelo de datos sabiendo que una tabla de calendario es una tabla de búsqueda no hay mayor misterio en ese punto y relacionarla con un campo fecha normalmente de la tabla de hechos de nuestro modelo.



1.4.2 Múltiples tablas de calendario en el mismo modelo

En un mismo modelo de datos **podemos tener más de una tabla de calendario cada uno con un propósito** por ejemplo, si hay varios campos de fecha en la tabla de hechos Ventas, como ClaveFecha FechaEnvío y FechaDevolución, todos pueden tener relaciones con el campo Fecha de la tabla de fechas Calendario, pero solo uno de ellos puede ser una relación activa. En este caso, puesto que ClaveFecha representa la fecha de la transacción y, por lo tanto, la fecha más importante, sería la relación activa más adecuada. Los demás tienen relaciones inactivas.

2 Sobre las funciones de inteligencia de tiempo

2.1 Presentación

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 ()

2.2 Creando funciones de inteligencia de tiempo con medidas rápidas.

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, tal y como vemos a continuación lo primero es seleccionar el cálculo 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.

Medida rápida

Cálculo
 Cambio de año a año
 Permite calcular el cambio año a año del valor base. [Más información](#)

Valor base
 Cantidad

Fecha
 Fecha

Número de períodos
 1

Campos
 Buscar

- Calendario
 - Fecha
- Cientes
- Productos
- Ventas
 - Cantidad
 - Cantidades
 - Diferencia Importes
 - Fecha
 - IdCliente
 - IdProducto
 - Importe vs Clientes
 - Importes
 - Importes Catálogo
 - Importes variables
 - Importes variables optimizado
 - Precio

De esta forma nos genera automáticamente la medida que podremos modificar o adaptar cambiándole el nombre y ubicación.

```

1 YoY% de Cantidad =
2 IF(
3   ISFILTERED('Calendario'[Fecha]),
4   ERROR("La medida rápida de inteligencia de tiempo solo se puede agrupar o filtrar mediante la jerarquía"),
5   VAR __PREV_YEAR = CALCULATE([Cantidad], DATEADD('Calendario'[Fecha].[Date], -1, YEAR))
6   RETURN
7     DIVIDE([Cantidad] - __PREV_YEAR, __PREV_YEAR)
8 )

```

Pero es más interesante y potente hacer nuestras medidas sin el asistente ya que nos permitirá usar más opciones y aprender el lenguaje DAX.

2.3 SAMEPERIODLASTYEAR. Compara importes en el mismo periodo

2.3.1 Concepto y sintaxis

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, por defecto será la columna fecha de nuestra dimensión de tiempo.

Sintaxis:

SAMEPERIODLASTYEAR (dates)

Argumentos de la función:

- dates: Columna conteniendo fechas.

2.3.2 Ejemplo: Ventas diarias del año pasado

Fuente: <https://inteligencia.dds.pe/dax/funciones-de-inteligencia-de-tiempo-en-dax/>

Vamos a crear 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.

SAMEPERIODLASTYEAR devuelve un conjunto de fechas en la selección actual, pero del año anterior. Por último, necesitamos hacer referencia a la columna **Fechas** de la tabla **Tiempo**.

```

Ventas del Año Pasado =
CALCULATE([Venta Total], SAMEPERIODLASTYEAR(Tiempo[Fecha]))

```

El resultado de la aplicación de esta medida con la función es el que se presenta en la siguiente tabla donde observamos los datos de ventas actuales en la columna **Venta total** con los de la columna de la medida **Ventas del Año Pasado**. La tabla **Ventas del Año Pasado** calcula las ventas totales, sin embargo, el contexto de la fecha se cambia al año anterior.

Fecha

01/01/2017 31/12/2018



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

2.3.3 Ejemplo: Ventas mensuales año anterior

Fuente: <https://interactivechaos.com/es/dax/function/sameperiodlastyear>

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 €	
		Total	3.144.393.292,13 €
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 €
		Total	2.642.413.217,03 €
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 €

2.4 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) ya que con TOTAL podemos especificar filtro opcionalmente y en el caso de DATE no es posible sino a través de funciones adicionales.

Por ejemplo, para analizar el acumulado anual de nuestras ventas tendríamos con:

DATESYTD

[Total Sales YTD] = CALCULATE([Total Sales];DATESYTD(Calendar[Date]))

TOTALYTD

Puede reescribirse como:

[Total Sales YTD] =TOTALYTD([Total Sales]; Calendar[Date])

2.4.1 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.

2.4.2 Ejemplo TOTALYTD

Fuente: <https://interactivechaos.com/es/dax/function/totalytd>

Disponemos de la siguiente tabla de ventas y definimos la medida de ventas totales como:

Ventas = SUM(Ventas[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:

Ventas YTD = TOTALYTD(Ventas[Ventas], Calendario[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

2.4.3 Ejemplos TOTALQTD

Fuente: <https://interactivechaos.com/es/dax/function/totalqtd>

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

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

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

2.4.4 Ejemplos TOTALMTD

Fuente: <https://interactivechaos.com/es/dax/function/totalmtd>

La medida "Ventas MTD" calcula el total de ventas desde principio del mes del contexto actual hasta la última fecha del contexto actual. Puede observarse cómo la medida se reinicializa a cero al principio de cada mes.

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

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

2.4.5 Ejemplo: DATESYTD junto a TOTALYTD

2.5 DATESYTD - DATESQTD- DATESMTD, evolucionadas a TOTALYTD - TOTALQTD - TOTALMTD

Estas funciones han evolucionado hacia TOTALYTD, TOTALQTD y TOTALMTD y por tanto su esencia es la misma que la explicada anteriormente para el caso de TOTAL radicando su diferencia como hemos comentado en que con TOTAL podemos especificar filtro opcionalmente y en el caso de DATE no es posible sino a través de funciones adicionales.

2.5.1 Concepto y sintaxis de estas funciones

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

Sintaxis:

DATESMTD(dates)

Argumentos de la función:

- dates: Columna conteniendo fechas.

2.5.2 Ejemplo DATESYTD

Fuente: <https://interactivechaos.com/es/dax/function/datesytd>

Veamos el siguiente caso, considere la siguiente tabla de ventas de los ejercicios 2015 y 2016 y definamos la siguiente medida

Ventas YTD = CALCULATE(Ventas[Ventas],DATESYTD(Calendario[Fecha]))

Si llevamos estas medidas a una matriz, obtenemos la siguiente tabla donde puede verse cómo la medida Ventas YTD acumula las ventas realizadas desde el 1 de enero de cada año.

Fecha	Importe	Mes Año	Ventas	Ventas YTD
<i>martes, 15 de septiembre de 2015</i>	2	2015-09	5	5
<i>miércoles, 30 de septiembre de 2015</i>	3	2015-10	3	8
<i>jueves, 1 de octubre de 2015</i>	1	2015-11	3	11
<i>miércoles, 7 de octubre de 2015</i>	2	2015-12		11
<i>miércoles, 11 de noviembre de 2015</i>	3	2016-01	7	7
<i>viernes, 1 de enero de 2016</i>	5	2016-02	6	13
<i>viernes, 15 de enero de 2016</i>	2	2016-03		13
<i>miércoles, 3 de febrero de 2016</i>	5	2016-04	2	15
<i>miércoles, 17 de febrero de 2016</i>	1	2016-05	3	18
<i>viernes, 1 de abril de 2016</i>	2	2016-06		18
<i>sábado, 21 de mayo de 2016</i>	3	2016-07	4	22
<i>viernes, 15 de julio de 2016</i>	4	2016-08		22
<i>miércoles, 28 de septiembre de 2016</i>	2	2016-09	2	24
		2016-10		24
		2016-11		24
		Total	35	24

Veamos otro caso con la siguiente medida similar a la anterior.

Total Sales YTD =CALCULATE([Total Sales], DATESYTD(DimDate[Datekey]))

CalendarMonthLabel ▲	Total Sales	Total Sales YTD
January	193.305.554,64 €	193.305.554,64 €
February	209.439.067,93 €	402.744.622,56 €
March	203.991.979,69 €	606.736.602,25 €
April	276.891.048,16 €	883.627.650,41 €
May	288.749.508,61 €	1.172.377.159,02 €
June	283.186.644,54 €	1.455.563.803,56 €
July	272.818.635,11 €	1.728.382.438,68 €
August	263.780.279,28 €	1.992.162.717,96 €
September	257.282.781,95 €	2.249.445.499,91 €
October	288.853.903,92 €	2.538.299.403,82 €
November	308.752.784,65 €	2.847.052.188,48 €
December	297.341.103,65 €	3.144.393.292,13 €
Total	3.144.393.292,13 €	3.144.393.292,13 €

2.5.3 Ejemplo DATESQTD

Fuente: <https://interactivechaos.com/es/dax/function/datesqtd>

Creamos la medida *Ventas QTD* que calcula las ventas acumuladas desde principio de cada trimestre:

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

Si llevamos estas medidas a una matriz, obtenemos el siguiente resultado de la matriz donde se observa cómo la medida se reinicializa a cero al principio de cada trimestre

Fecha	Importe	Mes Año	Ventas QTD
<i>martes, 15 de septiembre de 2015</i>	2	2015-09	5
<i>miércoles, 30 de septiembre de 2015</i>	3	2015-10	3
<i>lunes, 5 de octubre de 2015</i>	1	2015-11	6
<i>miércoles, 7 de octubre de 2015</i>	2	2015-12	6
<i>miércoles, 11 de noviembre de 2015</i>	3	2016-01	7
<i>viernes, 1 de enero de 2016</i>	5	2016-02	13
<i>viernes, 15 de enero de 2016</i>	2	2016-03	13
<i>miércoles, 3 de febrero de 2016</i>	5	2016-04	2
<i>miércoles, 17 de febrero de 2016</i>	1	2016-05	5
<i>viernes, 1 de abril de 2016</i>	2	2016-06	5
<i>sábado, 21 de mayo de 2016</i>	3	2016-07	4
<i>viernes, 15 de julio de 2016</i>	4	2016-08	4
<i>miércoles, 28 de septiembre de 2016</i>	2	2016-09	6
		Total	

2.5.4 Ejemplo DATESMTD

Fuente: <https://inteligencia.dds.pe/dax/funciones-de-inteligencia-de-tiempo-en-dax/>

En este ejemplo vamos a aplicar DATESMTD

$$\text{Ventas MTD} = \text{CALCULATE}([\text{Venta Total}], \text{DATESMTD}(\text{Tiempo}[\text{Fecha}]))$$

Cuando añadimos esta medida a una tabla con la medida *Venta Total*, se puede apreciar la diferencia de estas medidas.

Fecha	Venta Total	Ventas MTD
domingo, 29 de enero de 2017	\$1,282,219.56	\$19,798,123.13
unes, 30 de enero de 2017	\$1,515,557.70	\$21,313,680.83
martes, 31 de enero de 2017	\$1,783,692.25	\$23,097,373.08
miércoles, 1 de febrero de 2017	\$44,040.78	\$44,040.78
ueves, 2 de febrero de 2017	\$932,849.55	\$976,890.33
jueves, 23 de febrero de 2017	\$1,182,014.66	\$15,763,639.25
viernes, 24 de febrero de 2017	\$1,256,128.45	\$17,019,767.70
sábado, 25 de febrero de 2017	\$1,167,587.13	\$18,187,354.83
domingo, 26 de febrero de 2017	\$1,233,024.05	\$19,420,378.88
lunes, 27 de febrero de 2017	\$1,450,991.17	\$20,871,370.05
Total	\$808,639,703.54	\$33,579,854.88

Esta columna en realidad funciona casi como un total acumulado. Se reiniciará cada mes a medida que ingresen nuevos datos.

Para hacer un resultado parecido, pero no para el mes sino para el año o trimestre, solo necesita cambiar **DATESMTD** a **DATESYTD** o **DATESQTD**. A partir de ahí, puede realizar cálculos con sus datos de muchas otras formas.

2.6 DATEDIFF, para intervalos de fechas cálculo de la media de días.

2.6.1 Concepto y sintaxis

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**
- **MINUTE**
- **DAY**
- **WEEK**
- **MONTH**
- **QUARTER**
- **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.

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)

2.6.2 Ejemplo I

Fuente: <https://interactivechaos.com/es/dax/function/datediff>

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

The image displays eight screenshots of a Power BI table, each showing a different DATEDIFF calculation. The table columns are 'Fecha de inicio', 'Fecha final', and 'Diferencia'. The start date is 31/12/2015 23:59:59 and the end date is 01/01/2016 0:00:00.

Intervalo	Diferencia
SECOND	1
MINUTE	1
HOUR	1
DAY	1
WEEK	0
MONTH	1
QUARTER	1
YEAR	1

2.6.3 Ejemplo Semana desde la última venta.

Fuente: <https://interactivechaos.com/es/dax/function/datediff>

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):

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

2.7 DATEADD, por ejemplo para comparar con dos o tres años antes. Para calcular tasa de crecimiento.

2.7.1 Concepto y sintaxis

La función **DATEADD** devuelve una tabla con una columna de fechas que coinciden con las implicadas en el contexto actual desplazadas hacia adelante o hacia atrás un número de intervalos determinado. Es decir, 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.

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.

2.7.2 Ejemplo, comparar las ventas actuales con las de un periodo anterior concreto (por ejemplo dos años, tres, etc..)

Fuente: <https://inteligencia.dds.pe/dax/funciones-de-inteligencia-de-tiempo-en-dax/>

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 añadido 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
ueves, 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
Total	\$808,639,703.54	\$804,429,375.23

2.7.3 Ejemplo básicos I. Venta del mes vs Venta de mes anterior.

Crecimiento respecto a un periodo anterior.

Fuente: <https://interactivechaos.com/es/dax/function/dateadd>

Disponemos de una tabla con ventas y 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 €

2.7.4 Ejemplo, crecimiento respecto a un periodo anterior

Fuente: <https://interactivechaos.com/es/dax/scenario/crecimiento-respecto-un-periodo-anterior>

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

2.8 DATESBETWEEN

2.8.1 Concepto y sintaxis

La función **DATESBETWEEN** 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.

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.

2.8.2 Ejemplo Base

Fuente: <https://interactivechaos.com/es/dax/function/datesbetween>

Definimos la siguiente medida en la que calculamos las ventas entre el 2 de enero de 2007 y el 4 de enero de 2007:

```
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 €
martes, 2 de enero de 2007	6.270.657,17 €
miércoles, 3 de enero de 2007	6.096.024,11 €
jueves, 4 de enero de 2007	5.979.164,13 €
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 €
Total	8.341.224.364,83 €

18.345.845,41 €
Ventas período

Ventas período

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

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.

¡DATESBETWEEN() también funciona con las tablas desconectadas.

2.8.3 Ejemplo1: Tabla Dinámica con Medida Vida de Operación

Conociendo estas funciones podemos plantear un escenario interesante junto con DATESBETWEEN, consiste en evaluar una expresión desde el primer día de operación hasta la fecha presente en el contexto actual.

```
[Ingresos (Vida de Operación)]= CALCULATE ( [Ingresos]; DATESBETWEEN ( Calendario[Fecha]; FIRSDATE ( ALL ( Calendario[Fecha] ) ); LASTDATE ( Calendario[Fecha] ) ) )
```

Etiquetas de fila	Ingresos	Ingresos (Vida de Operación)
1999	\$8,685.83	\$8,686
2000	\$180,680.24	\$189,366
2001	\$166,860.36	\$356,226
2002	\$184,876.83	\$541,103
2003	\$177,061.95	\$718,165
2004	\$168,059.37	\$886,225
2005	\$187,906.44	\$1,074,131
2006	\$198,797.54	\$1,272,929
2007	\$170,712.79	\$1,443,641
2008	\$175,936.37	\$1,619,578
2009	\$198,222.01	\$1,817,800
2010	\$240,873.82	\$2,058,674
2011	\$216,088.04	\$2,274,762
2012	\$229,644.13	\$2,504,406
2013	\$232,463.65	\$2,736,869
2014	\$256,692.19	\$2,993,562
2015	\$327,323.99	\$3,320,886
Total general	\$3,320,885.54	\$3,320,886

fx
 fx
 fx
 fx
 fx
 fx
 fx
 fx

Arrastrar carr

FILTROS

FILAS

Año

Apilaz ar

Pedidos Descuento ... +

Promedio: \$161,957.77 Recuento: 10 Suma: \$1,619,577.71

2.8.4 Ejemplo 2

CalendarYear	Month - WeekNumOfYear	FIRSTDATE Example	LASTDATE Example
2001	January	1/1/2003	1/31/2003
2002	1	1/1/2003	1/4/2003
2003	2	1/5/2003	1/11/2003
2004	3	1/12/2003	1/18/2003
	4	1/19/2003	1/25/2003
	5	1/26/2003	1/31/2003
	February	2/1/2003	2/28/2003
	5	2/1/2003	2/1/2003
	6	2/2/2003	2/8/2003
	7	2/9/2003	2/15/2003
	8	2/16/2003	2/22/2003
	9	2/23/2003	2/28/2003
	March	3/1/2003	3/31/2003
	9	3/1/2003	3/1/2003

2.9 Otras funciones

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

- **TODAY**: Devuelve la fecha actual.
- **NOW**: Devuelve la fecha actual con horas, minutos y segundos.
- **FIRSTDATE**: 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 Anexos

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

1.5 Código de tabla calendario delimitado entre fechas

Fuente: [Olivia.es](#) - [Enlace al artículo](#)

```
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

1.6 Código de tabla calendario desde la fecha indicada hasta hoy

Fuente: clickam.es - [Enlace al artículo](#)

```

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")

```

1.7 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")

```

1.8 Formatos de fecha y hora definidos por el usuario

c	Muestra la fecha como dddd y muestra la hora como tttt, 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.
dddddd	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).
tttt	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.

4 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

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>

El ADN de Power Pivot. Miguel Caballero • Fabian Torres

Libro: Fórmulas DAX para Power Pivot. Una Guía Simple hacia la Revolución de Excel Rob Collie Holy Macro! Books PO Box 82 Uniontown, OH 44685 (Capítulo 14 pp 152 del pdf)