



DAX (III) Funciones de Inteligencia de tiempo

Funciones de Inteligencia de Tiempo

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

V.26

INDICE

1	Introducción a las funciones de inteligencia de tiempo	2
1.1	La tabla calendario como base de las funciones de inteligencia de tiempo	2
1.2	Clasificación de las funciones de inteligencia de tiempo	3
2	FIT Acumulativas: TOTAL y DATE (YTD,QTD,MTD).....	3
2.1	Presentación y principales usos.....	3
2.2	Tabla resumen comparativa	4
2.3	Ejemplos.....	4
2.3.1	Con TOTAL.....	4
2.3.2	Con DATE	6
2.4	Gráficos para el analisis acumulados respecto a periodos anteriores	8
3	FIT Comparativas: SAMEPERIODLASTYEAR, PARALLELPERIOD, DATEADD y PREVIOUS (YEAR, MONTH, QUARTER).....	9
3.1	Presentacion y principales usos.....	9
3.2	Tabla resumen comparativa	10
3.3	Comparativa DATEADD –SAMEPERIODLASTYEAR-PARALLELPERIOD	10
3.4	Ejemplos.....	11
3.4.1	Con SAMEPERODLASTYEAR.....	11
3.4.2	Con DATEADD	13
4	FIT de Intervalo: DATESBETWEEN y DATESINPERIOD	16
4.1	Presentación y principales usos.....	16
4.2	Tabla resumen comparativa	17
4.3	Ejemplos.....	18
4.3.1	Con DATESINPERIOD.....	18
4.3.2	Con DATESBETWEEN	19
5	FIT Otras funciones: TODAY, NOW, FIRSTDATE, LASTDATE,etc.....	22
6	DATEDIFF, para intervalos de fechas cálculo de la media de dias.....	22
6.1	Presentación y principales usos.....	22

6.2	Ejemplos.....	23
7	Referencias	24

1 Introducción a las funciones de inteligencia de tiempo

1.1 La tabla calendario como base de las funciones de inteligencia de tiempo

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)

1.2 Clasificación de las funciones de inteligencia de tiempo

Las funciones de Inteligencia de Tiempo (Time Intelligence) nos van a permitir resolver problemas utilizando el tiempo como nuestra dimensión o parámetro para hacer la comparación, permitiéndonos desplazarnos hacia el pasado (atrás) o hacia el futuro (delante)

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?
- Comparar la venta contra su mes previo o el año previo.
- ¿Cuál fue el mejor trimestre en los últimos dos años?
- Obtener acumulado mensual, trimestral o anual de ventas.

ACUMULATIVAS	COMPARATIVAS	DE INTERVALO	OTROS / DE VALOR ÚNICO
DATESYTD TOTALYTD	DATEADD SAMEPERIODLASTYEAR	DATESBETWEEN	FIRSTDATE LASTDATE
DATESQTD TOTALQTD	PARALLELPERIOD PREVIOUSDAY (YEAR, QUARTER, MONTH)	DATESINPERIOD	FIRSTNONBLANK FIRSTNONBLANKVALUE
DATESMTD TOTALMTD	NEXTDAY (YEAR, QUARTER, MONTH)		LASTNONBLANK LASTNONBLANKVALUE STARTOFYEAR (QUARTER/MONTH) ENDOFYEAR (QUARTER/MONTH)

2 FIT Acumulativas: TOTAL y DATE (YTD,QTD,MTD)

2.1 Presentación y principales usos

Permiten calcular valores acumulados sobre un período de tiempo, como días, meses, trimestres o años, respetando el contexto de filtros y jerarquías de fechas.

Calculan acumulados desde el inicio del período hasta la fecha actual en el contexto.

Se usan frecuentemente para crear indicadores como **Ventas acumuladas en el año (YTD)**, **Costos acumulados en el trimestre (QTD)**, etc.

Consideraciones **Las funciones TOTAL* devuelven un valor escalar (medida), mientras que las funciones DATES* devuelven una tabla de fechas.**

2.2 Tabla resumen comparativa

<i>Función - Descripción</i>	<i>Sintaxis - Ejemplo</i>
TOTALYTD Calcula el total acumulado desde el inicio del año hasta la fecha actual.	TOTALYTD(<expresión>, <columna_fecha>[, <filtro>][, <año_fin>]) Ventas YTD = TOTALYTD(SUM(Ventas[Monto]), 'Calendario'[Fecha])
TOTALQTD Acumulado desde el inicio del trimestre hasta la fecha actual.	TOTALQTD(<expresión>, <columna_fecha>[, <filtro>]) Ventas QTD = TOTALQTD(SUM(Ventas[Monto]), 'Calendario'[Fecha])
TOTALMTD Acumulado desde el inicio del mes hasta la fecha actual.	TOTALMTD(<expresión>, <columna_fecha>[, <filtro>]) Ventas MTD = TOTALMTD(SUM(Ventas[Monto]), 'Calendario'[Fecha])
DATESYTD Devuelve una tabla con todas las fechas desde el inicio del año hasta la actual.	DATESYTD(<columna_fecha>[, <año_fin>]) CALCULATE(SUM(Ventas[Monto]), DATESYTD('Calendario'[Fecha]))
DATESQTD Devuelve una tabla con todas las fechas desde el inicio del trimestre hasta la actual.	DATESQTD(<columna_fecha>) CALCULATE(SUM(Ventas[Monto]), DATESQTD('Calendario'[Fecha]))
DATESMTD Devuelve una tabla con todas las fechas desde el inicio del mes hasta la actual.	DATESMTD(<columna_fecha>) CALCULATE(SUM(Ventas[Monto]), DATESMTD('Calendario'[Fecha]))

Las funciones DATE 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.

2.3 Ejemplos

2.3.1 Con TOTAL

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

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

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

2.3.2 Con DATE

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

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

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.

$$\text{Total Sales YTD} = \text{CALCULATE}([\text{Total Sales}], \text{DATESYTD}(\text{DimDate}[\text{Datekey}]))$$

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 €

Ejemplo DATESQTD

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

Creemos 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	

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.

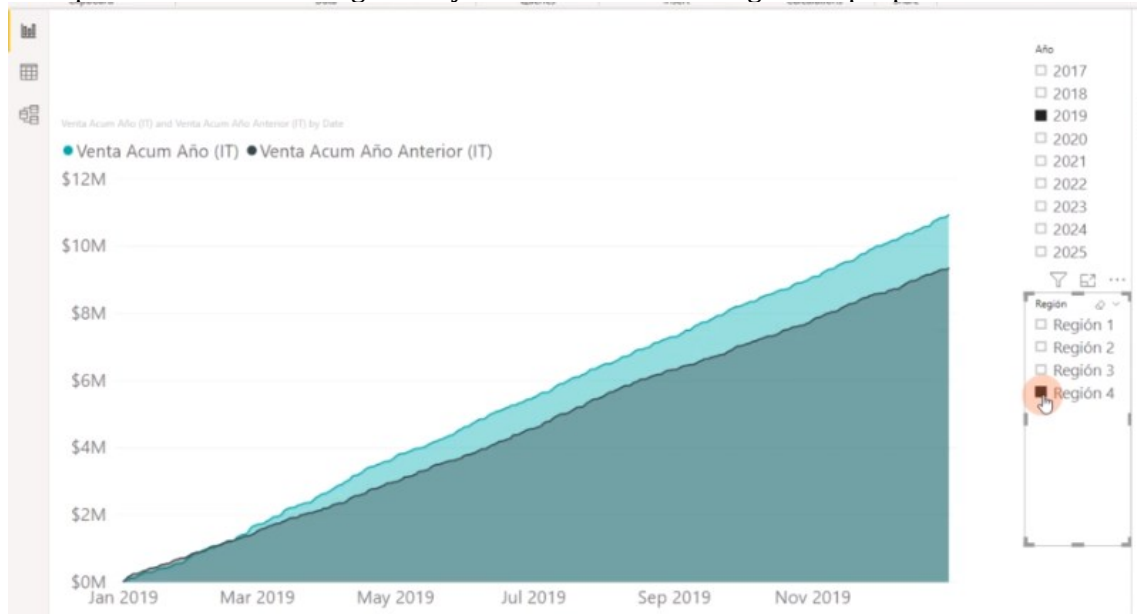
2.4 Gráficos para el análisis acumulados respecto a periodos anteriores

Curso Completo Power BI Desktop - Última versión de Power BI | Udemy - De **Diego Lopez**

Destacar que lo normal no es analizar estos acumulados a través de tablas sino con gráficos como el siguiente:



Este mismo análisis comparativo respecto al año anterior lo podríamos hacer incorporando la variable región tal y como tenemos en la siguiente propuesta.



3 FIT Comparativas: SAMEPERIODLASTYEAR, PARALLELPERIOD, DATEADD y PREVIOUS (YEAR, MONTH, QUARTER)

3.1 Presentacion y principales usos

Las funciones de inteligencia de tiempo comparativas son aquellas que permiten comparar valores de medidas entre diferentes períodos de tiempo, como el año anterior, el mes anterior o el mismo período en otro contexto.

Devuelven valores calculados en relación con un período distinto al actual (por ejemplo, comparar ventas actuales con las del año pasado).

Sus usos más frecuentes son:

- Comparar métricas año contra año (YoY), mes contra mes (MoM).
- Calcular variaciones porcentuales entre períodos.
- Analizar tendencias históricas.

Las más usadas son:

- SAMEPERIODLASTYEAR(), Devuelve el mismo período del año anterior. 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.
- PARALLELPERIOD(), Permite desplazar el período en intervalos (meses, trimestres, años).

- DATEADD(). Similar a PARALLELPERIOD, pero más flexible para desplazamientos positivos o negativos.
- PREVIOUSYEAR(), PREVIOUSMONTH(), PREVIOUSQUARTER(). Devuelven el período anterior específico

Se usan normalmente dentro de CALCULATE() para modificar el contexto de filtro.

Los intervalos válidos: DAY, MONTH, QUARTER, YEAR.

3.2 Tabla resumen comparativa

<i>Función - Descripción</i>	<i>Sintaxis - Ejemplo</i>
SAMEPERIODLASTYEAR() Devuelve el mismo período del año anterior.	SAMEPERIODLASTYEAR(<columnas_fecha> CALCULATE([Total Ventas], SAMEPERIODLASTYEAR('Calendario'[Fecha]))
PARALLELPERIOD() Devuelve el año anterior completo.	PARALLELPERIOD(<columnas_fecha>, <número>, <intervalo> CALCULATE([Total Ventas], PARALLELPERIOD('Calendario'[Fecha], -3, MONTH))
DATEADD() Desplaza el período en intervalos positivos o negativos.	DATEADD(<columnas_fecha>, <número>, <intervalo> CALCULATE([Total Ventas], DATEADD('Calendario'[Fecha], -1, MONTH))
PREVIOUSYEAR() Devuelve el año anterior completo.	PREVIOUSYEAR(<columnas_fecha>[, <número_de_años>]) CALCULATE([Total Ventas], PREVIOUSYEAR('Calendario'[Fecha]))
PREVIOUSMONTH() Devuelve el mes anterior completo.	PREVIOUSMONTH(<columnas_fecha> CALCULATE([Total Ventas], PREVIOUSMONTH('Calendario'[Fecha]))
PREVIOUSQUARTER() Devuelve el trimestre anterior completo.	PREVIOUSQUARTER(<columnas_fecha> CALCULATE([Total Ventas], PREVIOUSQUARTER('Calendario'[Fecha]))

Señalar que DateAdd comparte la característica de ser dinámica en el periodo como SamePeriodLastYear, pero SamePeriodLastYear solo dura un año en cambio DateAdd puede durar dos años o incluso más; **es por eso que DateAdd es una versión personalizada de SamePeriodLastYear.**

DateAdd no solo va atrás en el tiempo sino que también puede moverse hacia delante.

3.3 Comparativa DATEADD -SAMEPERIODLASTYEAR-PARALLELPERIOD

SAMEPERIODLASTYEAR y DATEADD (con parámetro “-1 year”) son equivalentes. Y devuelven el **mismo resultado**. SAMEPERIODLASTYEAR sólo sirve para año paralelo mientras que DATEADD permite agregar parámetros para calcular otros periodos paralelos: Meses, días...

1. DateAdd y SamePeriodLastYear son funciones que navegan en un periodo dinámico a través del contexto del filtro (Filter Context); con la diferencia que DateAdd puede retornar más de un año atrás.
2. DateAdd y ParallelPeriod pueden ir atrás y adelante según sea el parámetro y en cuanto a intervalo de tiempo DateAdd sirve para día, mes, trimestre y año, mientras que ParallelPeriod solo por mes, trimestre y año.

- ParallelPeriod a pesar de su gran parecido en estructura con DateAdd, el solo funciona de una manera estática en función de su intervalo

PARALLELPERIOD es similar a DATEADD. Pero la segunda permite especificar periodos paralelos hasta nivel día (“day“, ahora entiendes, no?). Mientras que la primera sólo admite “year”, “quarter”, “month” como tercer parámetro. Por eso, cuando has seleccionado del día 2/3/2018 al 15/5/2018, PARALLELPERIOD devuelve en este caso el importe del año anterior completo

3.4 Ejemplos

3.4.1 Con SAMEPERIODLASTYEAR

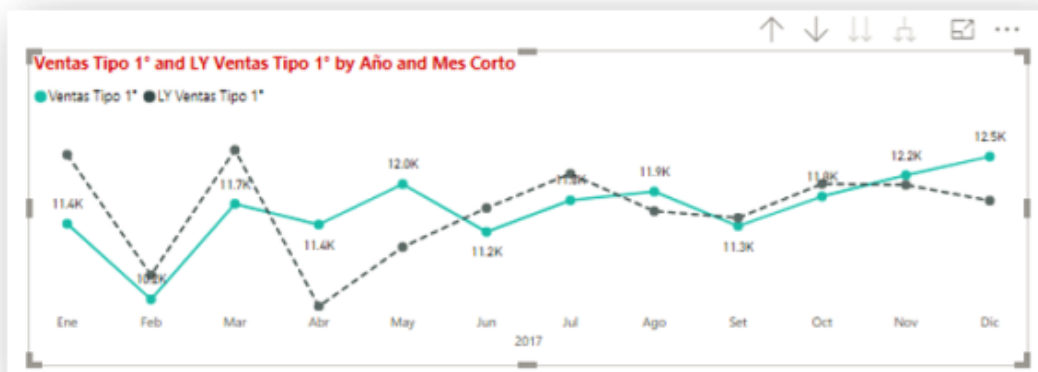
1 Venta Año Anterior = CALCULATE([Venta],
2 SAMEPERIODLASTYEAR('Tabla Calendario'[Date])

Año	Mes (nombre)	Venta	Venta Año Anterior
2019	ene	\$5,103,356	\$4,759,093
2019	feb	\$4,673,833	\$3,903,169
2019	mar	\$5,229,861	\$4,593,614
2019	abr	\$5,535,864	\$4,506,993
2019	may	\$5,517,240	\$5,191,316
2019	jun	\$4,779,318	\$4,968,405
2019	jul	\$5,822,736	\$5,218,068
2019	ago	\$4,810,312	\$4,958,490
2019	sep	\$4,923,560	\$4,142,190
2019	oct	\$5,280,771	\$4,799,075
2019	nov	\$4,829,296	\$5,143,098
2019	dic	\$4,864,979	\$4,582,172
2020	ene	\$5,294,788	\$5,103,356
2020	feb	\$4,383,889	\$4,673,833

Año	Venta	Venta Año Anterior
2019	\$61,368,127	\$56,765,683
2020	\$60,891,248	\$61,368,127
Total	\$122,259,375	\$118,133,811

Ventas vs Ventas LY

Año	Ventas	LY Ventas
2016	139,226.6	
Ene	12,486.4	
Feb	10,540.9	
Mar	12,562.0	
Abr	10,036.3	
May	10,994.9	
Jun	11,625.3	
Jul	12,178.7	
Ago	11,575.4	
Set	11,467.2	
Oct	12,015.6	
Nov	11,999.3	
Dic	11,744.8	
2017	139,229.1	139,226.6
Ene	11,372.0	12,486.4
Feb	10,150.8	10,540.9
Mar	11,692.8	12,562.0



Ejemplo: Ventas diarias del año pasado

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

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

Ejemplo: Ventas mensuales año anterior

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

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 €	3.144.393.292,13 €
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 €

3.4.2 Con DATEADD

3.4.2.1 Ejemplo mes anterior con DATEADD

En el siguiente ejemplo hemos calculado la suma de ventas del mes previo al seleccionado usando la función DATEADD, que lo que viene hacer es devolver una tabla con las fechas correspondientes al mes anterior del seleccionado y aplica la medida ventas (suma de ventas), es decir crear una tabla virtual con el conjunto de fechas correspondiente al mes anterior del que esta seleccionado o filtrado en este momento y sobre esta selección aplica la medida ventas.

```

1 Venta Mes Previo =
2 CALCULATE([Venta],
3     DATEADD('Tabla Calendario'[Date], -1, MONTH)
4 )
    
```


Año	Mes (nombre)	Venta	Venta Año Anterior	Venta Mes Previo
2019	ene	\$5,103,356	\$4,759,093	\$4,582,172
2019	feb	\$4,673,833	\$3,903,169	\$5,103,356
2019	mar	\$5,226,861	\$4,593,614	\$4,673,833
2019	abr	\$5,535,864	\$4,506,993	\$5,226,861
2019	may	\$5,517,240	\$5,191,316	\$5,535,864
2019	jun	\$4,779,318	\$4,968,405	\$5,517,240
2019	jul	\$5,822,736	\$5,218,068	\$4,779,318
2019	ago	\$4,810,312	\$4,958,490	\$5,822,736
2019	sep	\$4,923,560	\$4,142,190	\$4,810,312
2019	oct	\$5,280,771	\$4,799,075	\$4,923,560
2019	nov	\$4,829,296	\$5,143,098	\$5,280,771
2019	dic	\$4,864,979	\$4,582,172	\$4,829,296

3.4.2.2 Ejemplo trimestre anterior con DATEADD

```

1 Venta Trimestre previo =
2 CALCULATE([Venta],
3 | DATEADD('Tabla Calendario'[Date], -1, QUARTER)
4 )
    
```

Año Trimestre	Vtas Trimestre	Vtas Trimestre Año anterior	Vtas Trimestre anterior
2019 Trim. 1	\$15,004,051	\$13,255,876	\$14,524,345
2019 Trim. 2	\$15,832,422	\$14,666,715	\$15,004,051
2019 Trim. 3	\$15,556,608	\$14,318,748	\$15,832,422
2019 Trim. 4	\$14,975,046	\$14,524,345	\$15,556,608
2020 Trim. 1	\$14,722,178	\$15,004,051	\$14,975,046
2020 Trim. 2	\$16,045,740	\$15,832,422	\$14,722,178
2020 Trim. 3	\$15,110,860	\$15,556,608	\$16,045,740
2020 Trim. 4	\$15,012,469	\$14,975,046	\$15,110,860
Total	\$122,259,375	\$118,133,811	\$121,771,251

3.4.2.3 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

3.4.2.4 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 €

3.4.2.5 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:

$$[\% \text{ crecimiento}] = (\text{Valor actual} - \text{Valor hace un año}) / \text{Valor hace un año}$$

Para el cálculo del "Valor hace un año" vamos a utilizar la función DATEADD que devuelve una tabla conteniendo una columna de fechas que coinciden con las implicadas en el contexto actual desplazadas hacia adelante o hacia atrás un número de intervalos determinado. De esta forma, si la métrica es [Valor actual], podríamos definir:

$$[\text{Valor hace un año}] = \text{CALCULATE}([\text{Valor actual}]; \text{DATEADD}(\text{DimDate}[\text{Datekey}]; -1; \text{YEAR}))$$

Por último, la definición de la métrica buscada sería:

$$\% \text{ crecimiento} = \text{DIVIDE}(\text{[Valor actual]} - \text{[Valor hace un año]}; \text{[Valor hace un año]};)$$

La función DATEADD nos permite referir el crecimiento a cualquier número de días, meses, trimestres o años (atrás o adelante en el tiempo).

4 FIT de Intervalo: DATESBETWEEN y DATESINPERIOD

4.1 Presentación y principales usos

Las funciones de inteligencia de tiempo de intervalo se utilizan para devolver un conjunto de fechas dentro de un rango específico, lo que permite calcular medidas sobre períodos concretos. Las más comunes son: DATESBETWEEN y DATESINPERIOD

Funcionan de manera muy similar entre sí, pero tienen un uso un poco diferente; DatesInPeriod y DatesBetween y como primera aproximación tenemos

- DatesInPeriod es una función DAX perfecta para calcular períodos estándar que siguen intervalos de día, mes, trimestre y año **pero siempre en relación con el período actual**
- *DatesBetween* es una versión más genérica de DATESINPERIOD en el sentido de que ofrece más flexibilidad. Con esta función, no necesita preocuparse por el intervalo o el número de intervalos. Esta función le dará todas las fechas entre una fecha de

inicio y una fecha de finalización. Es útil cuando necesita analizar datos dentro de un intervalo de fechas específico.

La función **DATESINPERIOD** devuelve una tabla que contiene una columna de fechas que empieza por una fecha de inicio específica y sigue hasta el número y tipo de intervalos de fechas especificados, es decir, devuelve un conjunto de fechas a partir de un cierto día y cubriendo n intervalos del tipo indicado hacia atrás o hacia adelante en el tiempo..

Esta función es adecuada para pasar como filtro en la función CALCULATE, permite un control total sobre el intervalo de fechas de una medida.

```

1 Total Ventas Acumulado 3M =
2 //Total acumulado ultimos 3 meses
3 CALCULATE(
4     [Total Ventas],
5     DATESINPERIOD(
6         'Date'[Date],
7         MAX('Date'[Date]),
8         -3,
9         MONTH
10    )
11 )
    
```

Ilustración 1

Si el número de intervalos indicado es positivo, el intervalo indicado se añadirá a la fecha de inicio dicho número de veces. Por el contrario, si es negativo se restará. Pero siempre en relación con el período actual

4.2 Tabla resumen comparativa

<i>Función - Descripción</i>	<i>Sintaxis - Ejemplo</i>
<p>DATESBETWEEN</p> <p>Devuelve todas las fechas entre dos límites específicos (inclusive). Uso típico: Intervalos fijos: del 01/01 al 31/03.</p>	<p>DATESBETWEEN(<dates>, <start_date>, <end_date>)</p> <p>CALCULATE([Total Ventas], DATESBETWEEN('Calendario'[Fecha], DATE(2025,1,1), DATE(2025,3,31))) → Ventas entre el 1 Ene y 31 Mar 2025.</p>
<p>DATESINPERIOD</p> <p>Devuelve fechas en un período relativo a una fecha base (puede ser negativo para períodos previos). Uso típico: Últimos 12 meses, últimos 3 meses, etc.</p>	<p>DATESINPERIOD(<dates>, <start_date>, <number_of_intervals>, <interval>)</p> <p>CALCULATE([Total Ventas], DATESINPERIOD('Calendario'[Fecha], DATE(2025,3,31), -3, MONTH)) → Ventas últimos 3 meses hasta 31 Mar 2025.</p>

Diferencias clave

- *DATESBETWEEN* → Intervalo fijo entre dos fechas concretas.
- *DATESINPERIOD* → Intervalo dinámico relativo a una fecha base (ideal para cálculos móviles como últimos 12 meses)

4.3 Ejemplos

4.3.1 Con DATESINPERIOD

Ejemplo calculo del total de ventas anual móvil (Revenue MAT - Moving Annual Total) o Ingresos totales anuales móviles

Fuente: [La función DAX DATESINPERIOD | Soluciones G Com \(gcomsolutions.co.uk\)](#)

En este caso disponemos de una tabla con los ingresos totales según mes y año y en primer lugar vamos a calcular *Total Anual Móvil (MAT)* de los ingresos y que llamaremos a la medida como Revenue MAT.

Usaremos la función CALCULATE para la medida Ingresos Totales [Total Revenue] y como **argumento de filtro** usamos la función DATESINPERIOD, de esta forma la medida propuesta será:

Year	Month	Total Revenue
2014	January	£1,729,615
2014	February	£1,581,659
2014	March	£1,742,155
2014	April	£1,513,329
2014	May	£1,762,127

Ilustración 2

Year	Month	Total Revenue	Revenue MAT
2014	January	£1,729,615	£1,729,615
2014	February	£1,581,659	£3,311,274
2014	March	£1,742,155	£5,053,429
2014	April	£1,513,329	£6,566,758
2014	May	£1,762,127	£8,328,885
2014	June	£1,639,773	£9,968,658
2014	July	£1,609,794	£11,578,452
2014	August	£1,719,731	£13,298,183
2014	September	£1,635,975	£14,934,158
2014	October	£1,760,912	£16,695,070
2014	November	£1,796,654	£18,491,724
2014	December	£1,627,926	£20,119,650
2015	January	£1,837,315	£20,227,350
2015	February	£1,519,182	£20,164,872
2015	March	£1,810,628	£20,233,345
Total		£133,833,761	£26,884,906

Ilustración 3

El argumento Dates será 'Date Table'[Date], y para nuestro argumento Start Date, queremos la última fecha en el período actual que se puede obtener usando MAX('Date Table'[Date]).

Estamos trabajando hacia atrás a partir de ahí, por tanto, el número de intervalos es menos uno, y el intervalo, o unidades de tiempo, es el año.

Ejemplo cálculo del promedio de ingresos de los tres últimos meses

Fuente: [La función DAX DATESINPERIOD | Soluciones G Com \(gcomsolutions.co.uk\)](#)

Retomamos el caso anterior (¡Error! No se encuentra el origen de la referencia.) donde disponemos de una tabla con los ingresos totales según mes y año y nuestro objetivo es calcular el promedio mensual de ingresos de los tres últimos meses (la media móvil tres meses), lo que implica que tendremos que dividir el acumulado trimestral por tres y llamaremos a esta medida Revenue 3MMA

Por lo tanto, usamos la función DIVIDIR y dentro de eso, usamos CALCULATE. La expresión que se va a calcular es [Total Revenue] y, a continuación, para nuestra expresión de filtro, usamos DATESINPERIOD.

El argumento número de intervalos es menos tres; es decir, los últimos tres meses. Y el intervalo, o unidades, es Mes.

1	Revenue 3MMA =	Year	Month	Total Revenue	Revenue MAT	Revenue 3MMA
2	DIVIDE(2014	January	£1,729,615	£1,729,615	£576,538
3	CALCULATE(2014	February	£1,581,659	£3,311,274	£1,103,758
4	[Total Revenue],	2014	March	£1,742,155	£5,053,429	£1,684,476
5	DATESINPERIOD(2014	April	£1,513,329	£6,566,758	£1,612,381
6	'Date Table'[Date],	2014	May	£1,762,127	£8,328,885	£1,672,537
7	MAX('Date Table'[Date]),	2014	June	£1,639,773	£9,968,658	£1,638,410
8	-3,	2014	July	£1,609,794	£11,578,452	£1,670,565
9	MONTH)	2014	August	£1,719,731	£13,298,183	£1,656,432
10),	2014	September	£1,635,975	£14,934,158	£1,655,167
11	3	2014	October	£1,760,912	£16,695,070	£1,705,539
12)	2014	November	£1,796,654	£18,491,724	£1,731,180
		2014	December	£1,627,926	£20,119,650	£1,728,497
		2015	January	£1,837,315	£20,227,350	£1,753,965
		2015	February	£1,519,182	£20,164,872	£1,661,474
		2015	March	£1,810,628	£20,233,345	£1,722,375
		Total		£133,833,761	£26,884,906	£2,318,743

Ilustración 4

Cuando agregamos nuestra función Revenue 3MMA a nuestra tabla visual, podemos ver que, a partir del tercer mes, tenemos una cifra que podemos comparar con el rendimiento del mes actual para ver si el mes actual está por encima o por debajo de la media móvil de tres meses.

Ejemplo Calculo de ventas durante las dos semanas anteriores que terminan en una fecha dada

Fuente: [DATESINPERIOD | Interactive Chaos](#)

<p>En este ejemplo se calculan las ventas realizadas durante las dos semanas que terminan el 23 de abril de 2003 (la medida <i>Sales</i> suma la columna de importes de ventas)</p>	<pre>1 Fortnight sales = CALCULATE(2 [Sales]; 3 DATESINPERIOD('Calendar'[Date]; DATE(2003;4;23); -14; DAY) 4)</pre> <div style="text-align: center; border: 1px solid gray; padding: 10px; margin: 10px auto; width: fit-content;"> <p>220.036 €</p> <p>Fortnight sales</p> </div>
---	---

4.3.2 Con DATESBETWEEN

Ejemplo Calcular las ventas dentro de un intervalo de fechas específico

Fuente: [Una guía completa de las funciones DAX DATESBETWEEN y DATESINPERIOD | por Andrew Hubbard | Medio \(medium.com\)](#)

Consideremos un conjunto de registros que contiene los datos de ventas para una tienda minorista. Queremos calcular las ventas totales entre el 1 de enero de 2022 y el 30 de junio de 2022. La fórmula DAX que utiliza DATESBETWEEN sería:

Total Sales = CALCULATE(SUM(Sales[Amount]), DATESBETWEEN(Calendar[Date], DATE(2022, 1, 1), DATE(2022, 6, 30)))

En este ejemplo, usamos la función DATESBETWEEN para filtrar la columna Calendar[Date], que representa las fechas de nuestro conjunto de datos, en función del intervalo de fechas especificado. La función CALCULAR aplica el filtro a los datos de ventas y calcula la suma del importe de las ventas.

Otro Ejemplo

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.

Ejemplo Calculo de la cantidad de ventas del día actual y del anterior.

Fuente: [DAX Power BI: DATESBETWEEN & DATESINPERIOD | por Andrei Khaidarov, MVP de Microsoft, PhD | Plataforma de energía | Medio \(medium.com\)](#)

Vamos a contar la cantidad de ventas del día actual y del día anterior. Para obtener el valor anterior, podemos tomar la fecha y restar 1 de ella, así obtenemos la fecha anterior con la función FIRSTDATE y con la función LASTDATE obtendremos el último valor presente.

```

1 TotalCurrentAndPreviosDays =
2   CALCULATE(
3     SUM('Orders'[Total]),
4     DATESBETWEEN(
5       'Orders'[Date],
6       FIRSTDATE('Orders'[Date])-1,
7       LASTDATE('Orders'[Date])
8     )
9   )

```

Date	Total	TotalCurrentAndPreviosDays
31 октября 2021 г.	17 400	17 400
1 ноября 2021 г.	21 100	38 500
2 ноября 2021 г.	16 600	37 700
3 ноября 2021 г.	16 200	32 800
4 ноября 2021 г.	15 900	32 100

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

Ejemplo 2

CalendarYear	Month - WeekNumOfYear	FIRSDATE Example	LASTDATE Example
2001			
2002			
2003	January	1/1/2003	1/31/2003
	1	1/1/2003	1/4/2003
	2	1/5/2003	1/11/2003
	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

5 FIT Otras funciones: TODAY, NOW, FIRSTDATE, LASTDATE,etc

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.

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

6.1 Presentación y principales usos

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 calculo 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)

6.2 Ejemplos

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

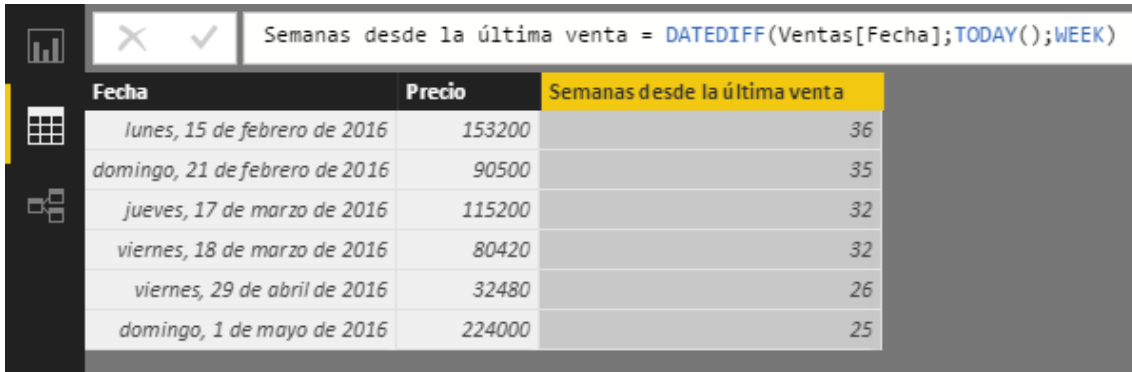
The image shows eight screenshots of the DAX DATEDIFF function being tested with different interval arguments. Each screenshot displays a table with the following columns: Fecha de inicio, Fecha final, and Diferencia. The start date is 31/12/2015 23:59:59 and the end date is 01/01/2016 0:00:00.

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

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):



The screenshot shows a Power BI interface with a table. The formula bar at the top displays the DAX formula: `Semanas desde la última venta = DATEDIFF(Ventas[Fecha];TODAY();WEEK)`. The table has three columns: 'Fecha', 'Precio', and 'Semanas desde la última venta'. The data rows are as follows:

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

7 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)

[Curso Completo Power BI Desktop - Última versión de Power BI | Udemy](#) - De **Diego**

Lopez

[DATEADD, PARALLELPERIOD y SAMEPERIODLASTYEAR – Business Intelligence and Analytics \(wordpress.com\)](#)