



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

Versión 2.5

V.26

- 1 **FIT Acumulativas: TOTAL y DATE (YTD,QTD,MTD)**
- 2 **FIT Comparativas: SAMEPERIODLASTYEAR, PARALLELPERIOD, DATEADD y PREVIOUS (YEAR, MONTH, QUARTER)**
- 3 **FIT de Intervalo: DATESBETWEEN y DATESINPERIOD**
- 4 **FIT Otras funciones: TODAY, NOW, FIRSDATE, LASTDATE,etc**
- 5 **DATEDIFF, para intervalos de fechas cálculo de la media de dias.**

1. 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 <i>TOTALYTD</i>	DATEADD <i>SAMEPERIODLASTYEAR</i>	DATESBETWEEN	FIRSTDATE LASTDATE
DATESQTD <i>TOTALQTD</i>	PARALLELPERIOD	DATESINPERIOD	FIRSTNONBLANK <i>FIRSTNONBLANKVALUE</i>
DATESMTD <i>TOTALMTD</i>	PREVIOUSDAY <i>(YEAR, QUARTER, MONTH)</i> NEXTDAY <i>(YEAR, QUARTER, MONTH)</i>		LASTNONBLANK <i>LASTNONBLANKVALUE</i>
			STARTOFYEAR <i>(QUARTER/MONTH)</i> ENDOFYEAR <i>(QUARTER/MONTH)</i>



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

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.

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

ACUMULATIVAS

DATESYTD
TOTALYTD

DATESQTD
TOTALQTD

DATESMTD
TOTALMTD

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.

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

Ejemplos

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

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

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

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

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

Ventas YTD \equiv **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
<i>martes, 15 de septiembre de 2015</i>	2
<i>miércoles, 30 de septiembre de 2015</i>	3
<i>jueves, 1 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	Ventas YTD
2015-09	5	5
2015-10	3	8
2015-11	3	11
2015-12		11
2016-01	7	7
2016-02	6	13
2016-03		13
2016-04	2	15
2016-05	3	18
2016-06		18
2016-07	4	22
2016-08		22
2016-09	2	24
2016-10		24
2016-11		24
Total	35	24

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

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. 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`.

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

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...

- 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.
- 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

Ejemplo con SAMEPERODLASTYEAR

```
1 Venta Año Anterior = CALCULATE([Venta],  
2 SAMEPERIODLASTYEAR('Tabla Calendario'[Date] )  
3
```

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	\$5,226,861	\$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

Filter pane (Values):

- Año
- Mes (nombre)
- Venta
- Venta Año Anterior

Legend (Año):

- 2017
- 2018
- 2019
- 2020
- 2021
- 2022
- 2023
- 2024
- 2025

Ejemplo con DATEADD

```
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

- Año
- 2017
 - 2018
 - 2019
 - 2020
 - 2021
 - 2022
 - 2023
 - 2024
 - 2025

```

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. FIT de Intervalo: DATESBETWEEN y DATESINPERIOD

Funcionan de manera muy similar, pero tienen un uso diferente; DatesInPeriod y DatesBetween

- 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* ofrece más flexibilidad. Es útil cuando necesita analizar datos dentro de un intervalo de fechas específico.

<i>Función - Descripción</i>	<i>Sintaxis - Ejemplo</i>
DATESBETWEEN Devuelve todas las fechas entre dos límites específicos (inclusive). Uso típico: Intervalos fijos: del 01/01 al 31/03.	DATESBETWEEN(<dates>, <start_date>, <end_date>) CALCULATE([Total Ventas], DATESBETWEEN('Calendario'[Fecha], DATE(2025,1,1), DATE(2025,3,31))) → Ventas entre el 1 Ene y 31 Mar 2025.
DATESINPERIOD 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.	DATESINPERIOD(<dates>, <start_date>, <number_of_intervals>, <interval>) CALCULATE([Total Ventas], DATESINPERIOD('Calendario'[Fecha], DATE(2025,3,31), -3, MONTH)) → Ventas últimos 3 meses hasta 31 Mar 2025.

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)

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

Ejemplo con DATESINPERIOD

```
1 Fortnight sales = CALCULATE(  
2     [Sales];  
3     DATESINPERIOD('Calendar'[Date]; DATE(2003;4;23); -14; DAY)  
4 )
```



En este ejemplo se calculan las ventas realizadas durante las dos semanas que terminan el 23 de abril de 2003 (la medida *Sales* suma la columna de importes de ventas)

Ejemplo con DATESBETWEEN

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

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)))

4. 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.

5 DATEDIFF, para intervalos de fechas cálculo de la media de días

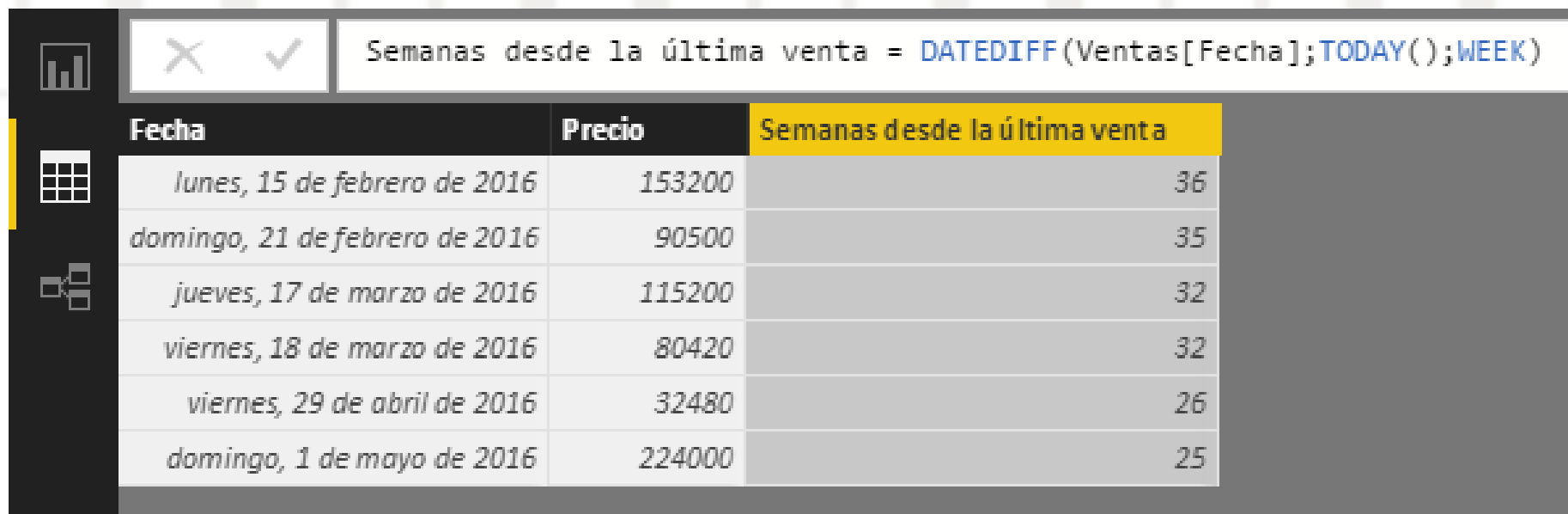
A través de esta función podemos averiguar el **intervalo de tiempo** que hay **entre dos fechas**. Los intervalos de tiempo que nos puede devolver esta función son los siguientes:

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

Ejemplo Semana desde la última venta

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\(\)](#)):

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



Semanas desde la última venta = `DATEDIFF(Ventas[Fecha];TODAY());WEEK)`

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