

Material Imprimible

Curso Power Bi Desktop

Módulo Inteligencia de Tiempo

Contenidos:

- ¿Qué es la inteligencia de tiempo?
- Tabla Calendario
- Funciones PreviousYear, PreviousMonth
- SAMEPERIODLASTYEAR
- Funciones de acumulación
- MoM, YoY

¿Qué es inteligencia de tiempo?

Son expresiones DAX que permiten manipular datos mediante períodos de tiempo (incluidos días, meses, trimestres y años) y, luego, compilar y comparar cálculos durante esos períodos.

Tabla Calendario

Para poder operar con la inteligencia de tiempo, es fundamental previamente haber creado una tabla calendario y declararla como tal. La principal característica de la tabla calendario es que contiene todas las fechas necesarias, consecutivas e ininterrumpidas. Es decir, contiene fechas de fin de semana, días hábiles y feriados, tenga transacciones o no.

Para crear la tabla calendario, existen dos funciones DAX que nos ayudan:

- CALENDARAUTO() : Esta función genera una tabla que recorre los datos de todo el modelo cargado, y genera una tabla con todas las fechas consecutivas entre la menor y la mayor fecha que encuentra, luego de analizar todos los campos fechas existentes.
- CALENDAR(startDate, endDate): Esta función permite generar la tabla calendario a partir de una fecha mínima y máxima que nosotros indiquemos. Podemos escribir estas fechas de forma manual o utilizar alguna función extra para complementar, como el TODAY() para detectar la fecha de "hoy".

También, se utilizan otras funciones DAX al crear la tabla calendario y extraer nuevas columnas para que la tabla calendario esté enriquecida.

En el siguiente bloque de fórmula observamos como se construye una tabla calendario con variados campos:

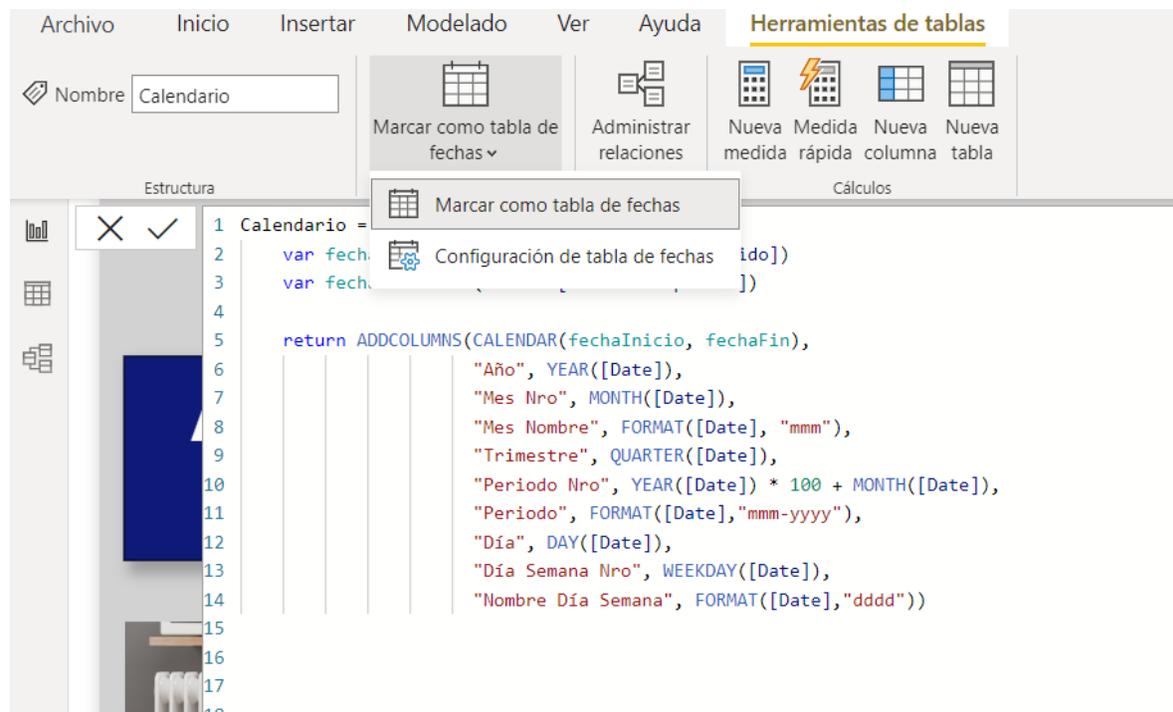
Calendario =

```
var fechaInicio = MIN(COMPRAS[Fecha del pedido])
var fechaFin = MAX(COMPRAS[Fecha del pedido])

return ADDCOLUMNS(CALENDAR(fechaInicio, fechaFin),
    "Año", YEAR([Date]),
    "Mes Nro", MONTH([Date]),
    "Mes Nombre", FORMAT([Date], "mmm"),
    "Trimestre", QUARTER([Date]),
    "Periodo Nro", YEAR([Date]) * 100 + MONTH([Date]),
    "Periodo", FORMAT([Date], "mmm-yyyy"),
    "Día", DAY([Date]),
    "Día Semana Nro", WEEKDAY([Date]),
    "Nombre Día Semana", FORMAT([Date], "dddd"))
```

En el mismo, para evitar introducir de forma manual las fechas, se usa el componente variable (var) para setear el mínimo y máximo de fechas de compras, y luego se retorna la tabla calendario con múltiples campos ya extraídos, que pueden ser muy útiles para mis análisis de tiempo futuros.

Antes de continuar a realizar cálculos es importante ir a “Modelado” teniendo seleccionada la tabla y declararla como “Tabla de Fechas”



El campo clave de esta tabla será “Date”.

Luego ya pudimos unir esta tabla con algún campo de fecha que tengamos en nuestro modelo de datos, como la Fecha de Venta o de Pedido.

Funciones PreviousYear y PreviousMonth

Previous Year: Devuelve una tabla que contiene una columna de todas las fechas del año anterior, en función de la primera fecha de la columna Dates del contexto actual. Sintaxis: PREVIOUSYEAR(Dates)

Previous Month: Devuelve una tabla que contiene una columna de todas las fechas del mes anterior, en función de la primera fecha de la columna Dates del contexto actual. PREVIOUSMONTH(Dates)

Se puede utilizar en cálculos que requieran interpretar una medida que ha ido evolucionando en el tiempo, por ejemplo:

CALCULATE(Sum(Ventas), PREVIOUSYEAR(Calendar[Date]))

De esta forma, utilizando la tabla calendario, tendremos las ventas del año anterior una vez que seleccionemos en algún filtro un año actual o visualicemos las ventas por años en algún gráfico que permita dicha comparación.

Función SamePeriodLastYear

Devuelve una tabla que contiene una columna de fechas desplazadas un año atrás en el tiempo desde las fechas de la columna dates especificada, en el contexto actual.

SAMEPERIODLASTYEAR(Dates)

Funciones de acumulación

DATESMTD:

Devuelve una tabla que contiene una columna con las fechas del mes hasta la fecha, en el contexto actual.

Su sintaxis es DATESMTD(Dates)

DATESYTD:

Devuelve una tabla que contiene una columna con las fechas del año hasta la fecha, en el contexto actual.

Su sintaxis es DATESYTD(Dates)

DATESQTD:

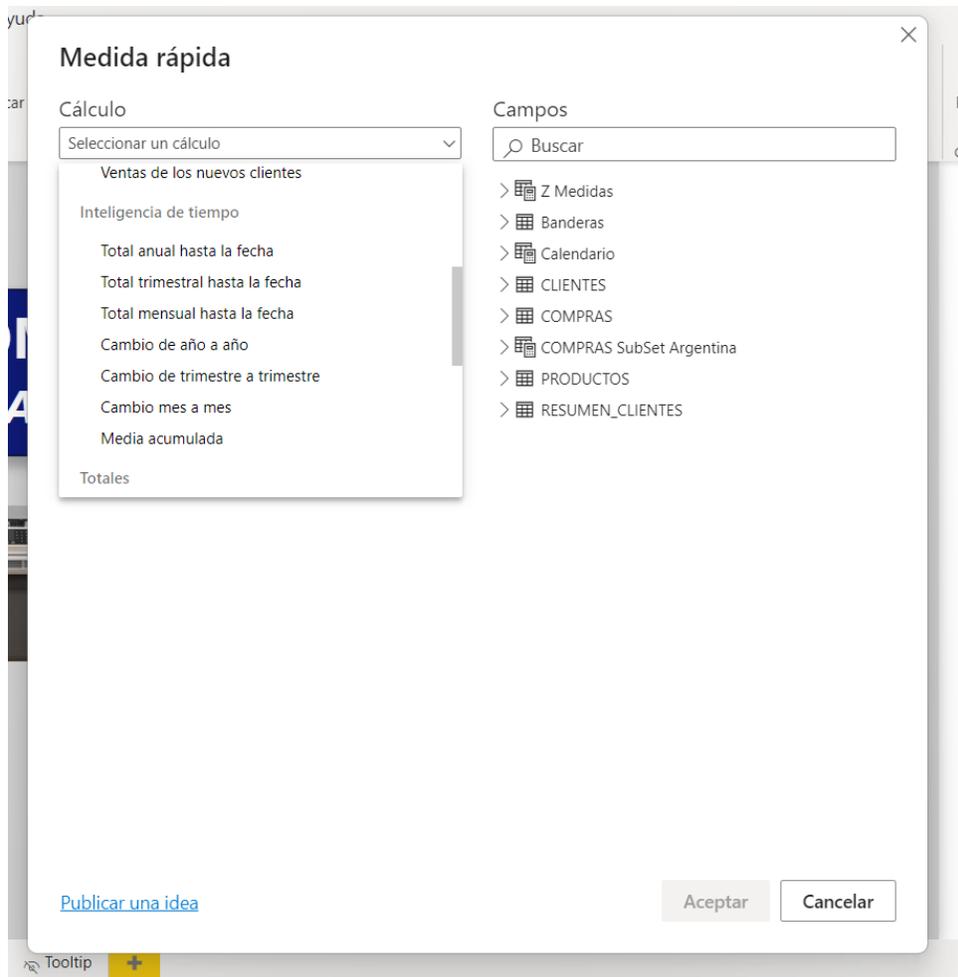
Devuelve una tabla que contiene una columna con las fechas del trimestre hasta la fecha, en el contexto actual.

Su sintaxis es DATESQTD(Dates)

MoM y YoY

Las variaciones temporales entre meses y años son muy populares en PowerBi, y se resuelve de varias formas. En el curso, utilizamos el creador de medidas rápidas DAX que tiene disponible estos cálculos de forma predefinida.

Debemos ir a **Medidas Rápidas > Inteligencia de tiempo**



En este caso, podremos realizar el cambio de año a año, y de mes a mes sin problemas. Solamente debemos introducir el campo "Date" de nuestra tabla calendario y la medida a comparar.

Medida rápida

Cálculo

Cambio de año a año ▼

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

Valor base ⓘ

Total Ventas ✕

Fecha ⓘ

Date ✕

Número de períodos ⓘ

1

Campos

🔍 Buscar

- 📊 # Total Ventas Año Anterior
- 📊 # Ventas a Individuos
- 📊 # Ventas Mes Anterior
- 📊 # Ventas Mobiliario
- 📊 # Ventas Mobiliario en Argentina
- 📊 # Ventas MTD
- 📊 # Ventas QTD
- 📊 # Ventas Totales Fijadas
- 📊 # Ventas YTD
- 📊 % Variacion Año Ant
- Σ Columna
- 📊 MoM% de # Total Ventas
- 📊 YoY% de # Total Ventas
- 📊 z Ratio Mobiliario Fijo
- 📊 z Ratio Mobiliario Total

> 🏳️ Banderas

✓ 📅 Calendario

Año

📅 Date

Día

Día Semana Nro

Mes Nombre

[Publicar una idea](#)

Aceptar

Cancelar

Esto generará una fórmula escrita en DAX que calculará la variación en una medida que podremos utilizar en los distintos gráficos.

MoM% de # Total Ventas =

```
VAR __PREV_MONTH = CALCULATE([# Total Ventas], DATEADD('Calendario'[Date], -1, MONTH))
```

```
RETURN
```

```
    DIVIDE([# Total Ventas] - __PREV_MONTH, __PREV_MONTH)
```

Podríamos escribir la fórmula directamente en código DAX, pero es una buena alternativa utilizar la interfaz de medidas rápidas, dado la complejidad de usar más de 2 funciones en simultáneo.