

1 Excel y el lenguaje DAX, aspectos generales

- 1.1 Excel, el modelo tabular y el lenguaje Excel. Diferencia de DAX y el lenguaje de fórmulas Excel
- 1.2 Tipos de datos en DAX según resultado
- 1.3 El entorno de trabajo de DAX en Power BI y Power Pivot

2 Sobre las funciones DAX, características y sintaxis

- 2.1 Categorías de las funciones DAX
- 2.2 Creación de medidas rápidas con el asistente y el administrador de medidas
- 2.3 Consejos relacionados con la creación de medidas y columnas calculadas
- 2.4 Aprendiendo a entender las funciones DAX
- 2.5 Entendiendo lo que son los iteradores. Funciones Sufijo X

3 Principales funciones DAX

- 3.1 Introducción
- 3.2 Funciones de agregación: SUM, MAX, PRODUCT, COUNT, etc
- 3.3 Funciones de Conteo
- 3.4 Funciones Lógicas
- 3.5 Funciones fecha y hora sin Time Intelligence
- 3.6 Funciones de información
- 3.7 Funciones de Texto
- 3.8 Funciones de navegación entre tablas. RELATED y RELATEDTABLET

4 Consejos y consideraciones a tener en cuenta en las fórmulas DAX

- 4.1 Evitar obtener un error como resultado de una fórmula
- 4.2 Errores Matemáticos
- 4.3 Claridad, Reglas para la presentación de fórmulas DAX óptima

5 Ejemplos de empleo de funciones DAX frecuentes

- 5.1 Uso de la función COUNTROWS y con filtro
- 5.2 Uso de la función Mediana
- 5.3 Uso de las funciones COUNTROWS() y DISTINCTCOUNT(). Calcular el promedio de ventas y las ventas medias diarias



JOSE IGNACIO GONZALEZ ...
jggomez@ull.edu.es



1. Excel y el lenguaje DAX, aspectos generales

A continuación, presentamos una visión general de la potencia del lenguaje y como este refuerza el modelo tabular como base de nuestros informes y cuadros de mando para lo cual nos apoyaremos en el uso del complemento Power BI y en la creación, conocimiento y comprensión de las medidas rápidas y el código autogenerado DAX

1.1 Excel, el modelo tabular y el lenguaje Excel . Diferencia de DAX y el lenguaje de fórmulas Excel

El lenguaje DAX se creó a partir del lenguaje Excel y por tanto incluye todas las funciones Excel y otras propias. Este lenguaje se basa en consultas a los modelos tabulares presente en el Power Pivot integrado en Excel, así como en Power BI en SQL y otros.

DAX es un lenguaje de funciones orientado al análisis de datos y puede ser considerado también como un lenguaje de consulta.

El conocimiento básico de modelo de bases de datos relacionales como SQL o Access facilita la comprensión de los modelos tabulares que recordemos se basan en tablas y sus relaciones y las funciones DAX tienen en cuenta esta estructura

Características generales del lenguaje DAX:

- Una gran parte de las funciones de DAX se denominan igual que las de Excel y tienen el mismo comportamiento, solo cambia la referencia, así en el caso de DAX es una columna o tabla de referencia mientras que en Excel son una celda o rango de celdas.
- Con DAX podemos crear columnas, campos (medidas) y tablas.
- DAX cuenta con una tipología de funciones propias que permite devolver una tabla de valores o realizar cálculos sobre una tabla de valores como entrada y que no está disponible en Excel.

Principales diferencias entre el lenguaje DAX y las fórmulas de Excel son:

- En DAX no existe el concepto celda de Excel. DAX trabaja sobre tablas, las funciones esperan tablas o columnas
- Una función de DAX siempre hace referencia a una columna o una tabla. Si solo desea usar valores concretos de una tabla o columna, puede agregar filtros a la fórmula.
- DAX aporta nuevas funciones que fortalecen el potencial de Excel como son: Funciones de inteligencia de tiempo e Iteradores.

1.2 Tipos de datos en DAX según resultado

Atendiendo al tipo de valor que arrojan las fórmulas o funciones DAX podemos clasificarlas en dos tipos:

Funciones DAX Escalares: Función o fórmula que arroja como resultado final una escalar, es decir un valor único, solo un elemento, ejemplo:

= SUM (TABLA_Ventas[Precio de Venta])

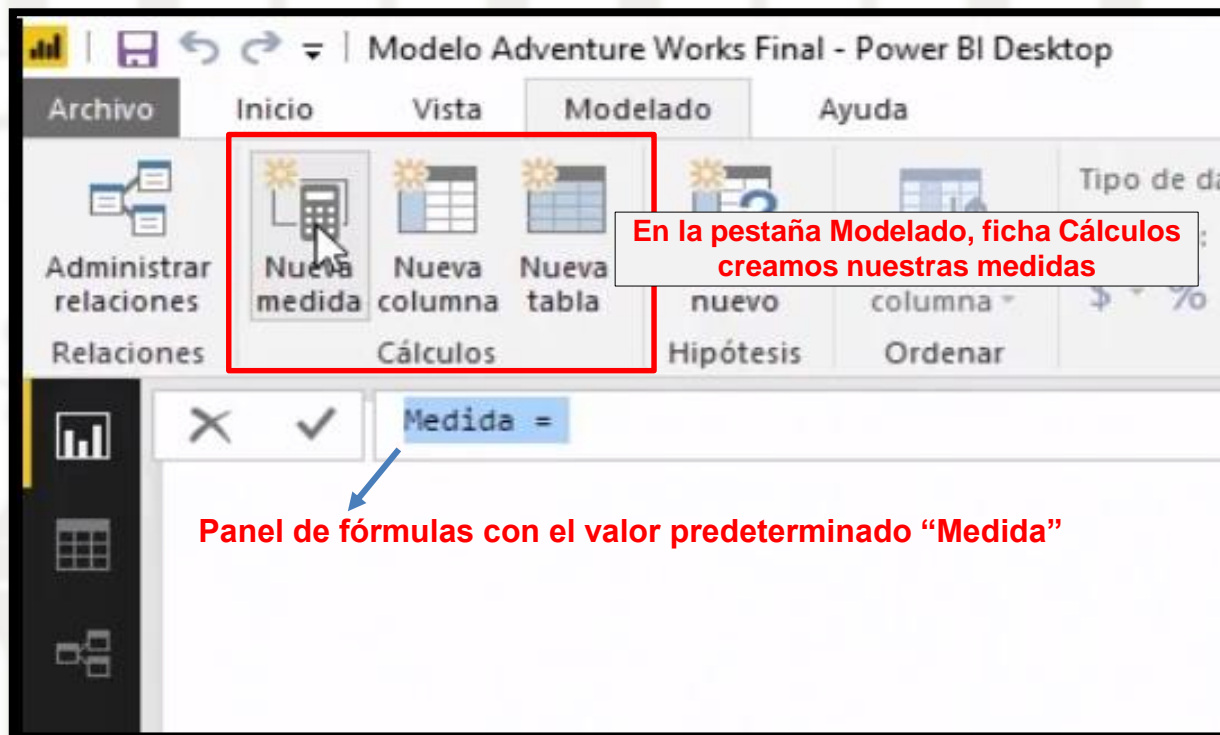
Funciones de Tabla: fórmula que devuelve como resultado una tabla o un conjunto de resultados. Ejemplo la siguiente expresión devuelve un subconjunto de la tabla Ventas de tal manera que únicamente devuelve los registros que son ≥ 80 € en el campo Precio de Venta:

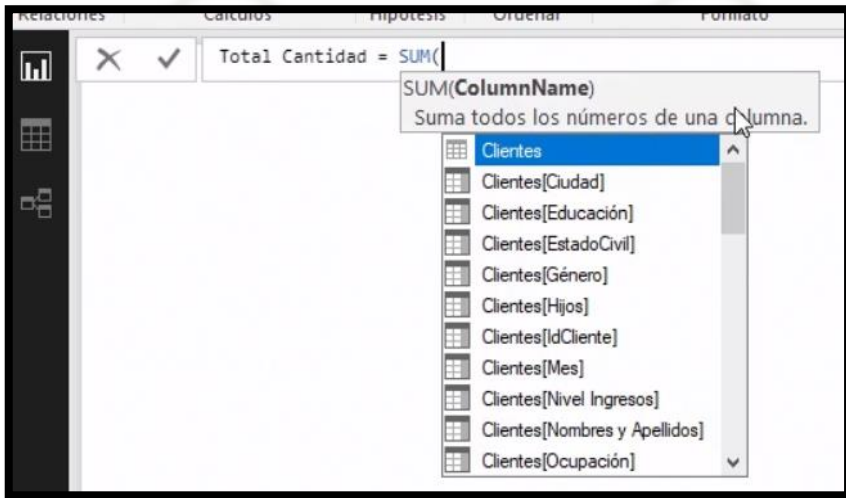
FILTER (TABLA_Ventas; TABLA_Ventas[Precio de Venta] \geq 80)

1.3 El entorno de trabajo de DAX en Power BI y Power Pivot

En Power BI disponemos de un área para crear las medidas, en concreto en la Ficha Modelado, Grupo Cálculos

Entorno de trabajo en Power BI. La ficha de modelado



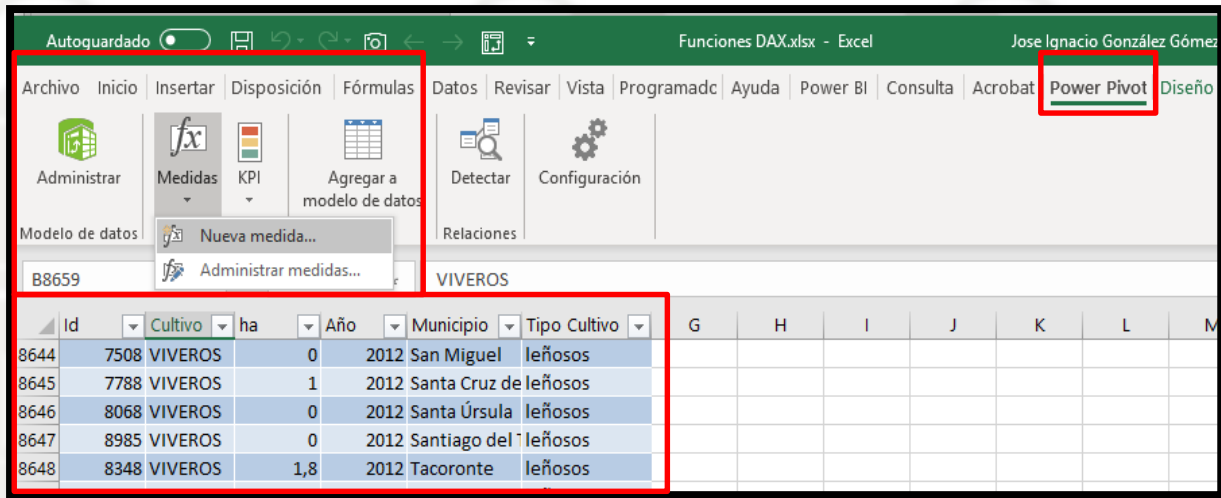


Quando estamos creando una medida una vez seleccionada la función, por ejemplo, SUM a continuación DAX muestra una ayuda que informa sobre los parámetros que admite cada función, y sobre todo nos avisa sobre el tipo de datos que está esperando (ColumnName), en el ejemplo la función SUM solo admite un nombre de columna.

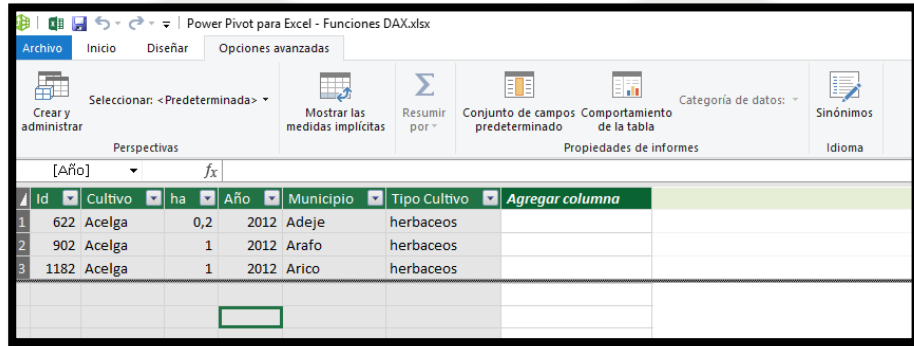
Se muestra también una lista con las tablas y columnas del modelo de datos que podemos seleccionar. El nombre de la columna esta precedido del nombre de la tabla.

Entorno de trabajo en Excel. Acceso a la pestaña Power Pivot y Administrar Modelo de Datos

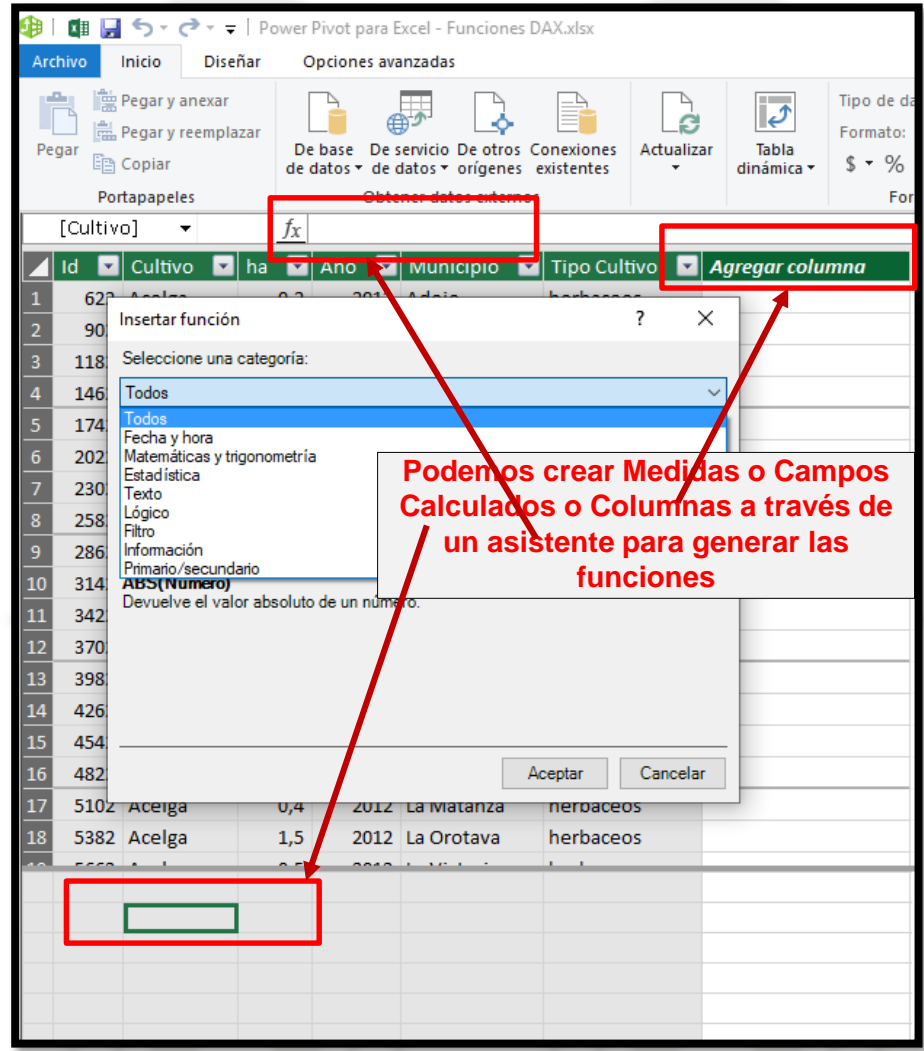
A través de la opción Power Pivot accedemos al centro de operaciones para la creación de medidas, KPI, Agregar Tablas a Modelo de Datos, etc.



Con la opción Administrar accedemos a una nueva pantalla que es el corazón del modelo de datos, disponiendo de las siguientes opciones principales agrupadas en tres pestañas:



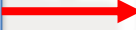
- Obtener datos externos y agregar al modelo de datos
- Establecer y administrar conexiones a datos y vincularla al modelo de datos
- Formateo de los datos
- Configurar y establecer las relaciones entre las fuentes de datos
- Crear y gestionar los campos calculados (medidas) y columnas calculadas.
- Crear tablas y gráficos dinámicos
- Crear KPI y otros cálculos



2. Sobre las funciones DAX, características y sintaxis

2.1 Categorías de las funciones DAX

Las funciones DAX se encuentran agrupadas en categorías y que hemos relacionado en grupos afines

- a) Funciones de Agregación
 - b) Funciones de Conteo
 - c) Funciones Lógicas
 - d) Funciones de Fecha y Hora
 - e) Funciones de Información
 - f) Funciones de Texto
- 

1. Fecha y Hora
2. Matemáticas y Trigonométricas
3. Estadísticas
4. Texto
5. Lógicas
6. Filtro
7. Información
8. Primario / Secundario
9. **Time Intelligence**
10. **Misceláneas**

La creación de medidas se fundamenta en la aplicación de alguna de las funciones anteriores y en su correcta parametrización.

Vamos a centrarnos a continuación en los aspectos comunes relacionados con todo este conjunto de funciones vinculados con sus reglas y sintaxis

2.2 Creación de medidas rápidas con el asistente y el administrador de medidas

Disponemos de un asistente tanto en Power BI como Power Pivot que nos facilita crear medidas rápidas generando el código automáticamente lo que al principio es de gran ayuda, pero ***es importante saber leer y comprender el código generado de forma automática***

Creación de medidas rápidas en Power BI

Las medidas rápidas son expresiones de cálculo escritas en DAX que vienen integradas en Power BI y que son de gran utilidad en el proceso de aprendizaje de DAX ya que funcionan con un asistente intuitivo que nos permite programar la medida de forma visual sin usar el lenguaje, es decir, el asistente generara de forma automáticamente el código DAX

Veamos el siguiente caso, contamos con un informe que contiene una sola columna que relaciona el total de ventas (cantidad) según país (fila), informe que está basado en un modelo de datos (activar vista de relaciones) compuesto por 4 tablas.

Contamos con un informe basado en un modelo de datos

País	Total Cantidad
Estados Unidos	21.344
Australia	13.345
Canada	7.620
Reino Unido	6.906
Alemania	5.625
Francia	5.558
Total	60.398

The data model consists of the following tables and fields:

- Clientes:** IdCliente, Nombres y Apellidos, EstadoCivil, Género, Hijos, Educación, Ocupación, Mes, OrdenMes, Nivel Ingresos, OrdenNivelIngresos, País, Ciudad
- Ventas:** CodProducto, CodCliente, Cantidad, Precio, Coste, Venta, %IVA, IVA, Fecha Orden, Total Cantidad
- Productos:** IdProducto, Producto, Color, PrecioCatálogo, Tamaño, RangoTamaño, Línea, Modelo, Subcategoría, Categoría, Veces Vendido, Importe Ventas, Primera Venta
- Fechas:** Fecha

Relationships are shown between Clientes and Ventas, and between Ventas and Productos.

Objetivo: Contar con una medida (columna en el informe) que nos de la diferencia entre EEUU y cada uno de los países

Nuestro objetivo es crear una medida que nos permita obtener una columna que muestre la diferencia entre la cantidad de EEUU y de cada uno de los países de la fila visibles.

Esta medida la podemos crear usando el lenguaje DAX escribiendo las expresiones o bien a través de la opción de medidas rápidas comentada y disponible en Power BI

Para acceder a la opción de medidas rápidas debemos acceder a la visualización informe o vista informe, panel de visualizaciones y en vista de campos vemos que en la propiedad valores tenemos los campos país y total cantidad, que son los campos disponibles y en el lateral derecho hay un pequeño triángulo y pulsando aparece un menú contextual que nos permite entre otras opciones la de crear una “Nueva medida rápida” y a través de la cual accedemos a un asistente para configurarla.

The screenshot shows the Power BI interface with a table of countries and their total quantities. A context menu is open over the 'Total Cantidad' field, with 'Nueva medida rápida' highlighted. A red arrow points from this option to the 'Medida rápida' dialog box.

País	Total Cantidad
Estados Unidos	21.344
Australia	13.345
Canada	7.620
Reino Unido	6.906
Alemania	5.625
Francia	5.558
Total	

Context menu options for 'Total Cantidad':

- Quitar campo
- Cambiar nombre
- Formato condicional
- Quitar el formato condicional
- Mostrar valor como
- Nueva medida rápida**

Medida rápida dialog box:

- Cálculo: Seleccionar un cálculo
- Campos: Buscar
- Clientes
- Fechas
- Productos
- Ventas

Una vez hemos accedido al asistente lo primero es seleccionar la opción cálculo para definir o seleccionar el tipo de cálculo deseado.

Destaca que nos pone a disposición un gran número de opciones de cálculo agrupadas o clasificadas en grupos o categorías en nuestro caso particular nos interesa ir al grupo filtros y seleccionar la opción “Diferencia respecto al valor filtrado” y de esta forma accedemos a configurarlo.

Medida rápida

Cálculo

Seleccionar un cálculo

Seleccionar un cálculo

Agregado por categoría

- Promedio por categoría
- Varianza por categoría
- Valor máximo por categoría
- Valor mínimo por categoría
- Media ponderada por categoría

Filtros

- Valor filtrado
- Diferencia respecto al valor filtrado**
- Diferencia porcentual respecto al valor filtrado
- Ventas de los nuevos clientes

Inteligencia de tiempo

- Total anual hasta la fecha
- Total trimestral hasta la fecha
- Total mensual hasta la fecha
- Cambio de año a año
- Cambio de trimestre a trimestre
- Cambio mes a mes
- Media acumulada

Campos

Buscar

- Clientes
- Fechas
- Productos
- Ventas

Medida rápida

Cálculo

Seleccionar un cálculo

Seleccionar un cálculo

Agregado por categoría

- Promedio por categoría
- Varianza por categoría
- Valor máximo por categoría
- Valor mínimo por categoría
- Media ponderada por categoría

Filtros

- Valor filtrado
- Diferencia respecto al valor filtrado
- Diferencia porcentual respecto al valor filtrado
- Ventas de los nuevos clientes

Inteligencia de tiempo

- Total anual hasta la fecha
- Total trimestral hasta la fecha
- Total mensual hasta la fecha
- Cambio de año a año
- Cambio de trimestre a trimestre
- Cambio mes a mes
- Media acumulada

Totales

- Total acumulado
- Total de la categoría (filtros aplicados)
- Total de la categoría (filtros no aplicados)

Operaciones matemáticas

- Suma
- Resta
- Multiplicación
- División
- Diferencia porcentual
- Coefficiente de correlación

Texto

- Clasificación por estrellas
- Lista concatenada de valores

Por tanto, configuramos los valores con los parámetros básicos, tal y como se muestra en la ilustración, seleccionando el valor base como total cantidad que es lo que estamos viendo en la tabla y es la medida que queremos comparar.

En el cuadro espacios en blanco podemos definir si lo que queremos es que se generen espacios en blanco en el resultado por ejemplo en caso de que exista un país que no tenga ventas o que se trate el dato como 0 y lo que vamos a comparar es las ventas de ese país igual a 0 contra las ventas en EEUU lo dejamos en generar espacio.

País	Total Cantidad
Estados Unidos	21.344
Australia	13.345
Canada	7.620
Reino Unido	6.906
Alemania	5.625
Francia	5.558
Total	60.398

Para el filtro ya PBI ha escogido país que es la columna que estamos utilizando la tablas la columna y sólo nos queda seleccionar el valor en este caso seleccionar EE.UU que es el país que queremos utilizar como referencia.

Ya todo configurado aceptamos aceptar y surge la magia, Power BI ha crea por nosotros el código DAX necesario para calcular la diferencia entre EE.UU y el resto de los países agregando la medida a la tabla y además al modelo, así como nos muestra el código generado para crear la medida.

```
Diferencia entre Total Cantidad y Estados Unidos =  
VAR __BASELINE_VALUE = CALCULATE([Total Cantidad]; 'Clientes'[País] IN { "Estados Unidos" })  
VAR __MEASURE_VALUE = [Total Cantidad]  
RETURN  
IF(NOT ISBLANK(__MEASURE_VALUE); __MEASURE_VALUE - __BASELINE_VALUE)
```

País		
Estados Unidos		
Australia	13.345	-7.999
Canada	7.620	-13.724
Reino Unido	6.906	-14.438
Alemania	5.625	-15.719
Francia	5.558	-15.786
Total	60.398	39.054

Medida rápida

Cálculo

Seleccionar un cálculo

- Media ponderada por categoría
- Filtros**
 - Valor filtrado
 - Diferencia respecto al valor filtrado
 - Diferencia porcentual respecto al valor filtrado
 - Ventas de los nuevos clientes
- Inteligencia de tiempo**
 - Total anual hasta la fecha
 - Total trimestral hasta la fecha
 - Total mensual hasta la fecha
 - Cambio de año a año**
 - Cambio de trimestre a trimestre
 - Cambio mes a mes
 - Media acumulada
- Totales**
 - Total acumulado
 - Total de la categoría (filtros aplicados)
 - Total de la categoría (filtros no aplicados)
- Operaciones matemáticas**
 - Suma

Campos

Buscar

- Calendario
- Cientes
- Productos
- Ventas

Medida rápida

Cálculo

Cambio de año a año

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

Valor base ⓘ

Total Cantidad

Fecha ⓘ

Fecha

Número de períodos ⓘ

1

Campos

Buscar

- Calendario
- Cientes
- Productos
- Ventas

YoY% de Total Cantidad =

```

IF(
    ISFILTERED('Calendario'[Fecha]);
    ERROR("La medida rápida de inteligencia de tiempo solo se puede agrupar o filtrar mediante la jerarquía de datos proporcionada por Power BI o por la columna de datos principal.");
    VAR __PREV_YEAR = CALCULATE([Total Cantidad]; DATEADD('Calendario'[Fecha].[Date]; -1; YEAR))
    RETURN
        DIVIDE([Total Cantidad] - __PREV_YEAR; __PREV_YEAR)
)
  
```

Año	Total	YoY%
2005		
Trim. 3	448	
Trim. 4	565	
2006	2.677	1013
Trim. 1	558	
Trim. 2	635	
Trim. 3	732	448
Trim. 4	752	565
2007	24.443	2677
		813,07 %

The image shows a Power BI report interface. On the left, a table displays sales data by country. The table has three columns: 'País', 'Total Cantidad', and 'Diferencia Total Cantidad y EEUU'. The data is as follows:

País	Total Cantidad	Diferencia Total Cantidad y EEUU
Estados Unidos	21.344	0
Australia	13.345	-7.999
Canada	7.620	-13.724
Reino Unido	6.906	-14.438
Alemania	5.625	-15.719
Francia	5.558	-15.786
Total	60.398	39.054

On the right, the 'VISUALIZACIONES' sidebar shows various chart types. Below it, the 'Valores' section lists the fields used in the table: 'País', 'Total Cantidad', and 'Diferencia Total Cantida'. The 'FILTROS' section shows filters for 'Diferencia Total Cantidad ...'. The 'CAMPOS' sidebar shows a list of fields, with 'Diferencia Tot...' selected.

El código siempre va a tener las funciones y expresiones que operan de la forma más eficiente para el rendimiento del informe. El código además hace uso de las variables en DAX son enormemente útiles y eficientes no debemos ignorar este código escrito por PBI porque representa un excelente recurso de aprendizaje del lenguaje.

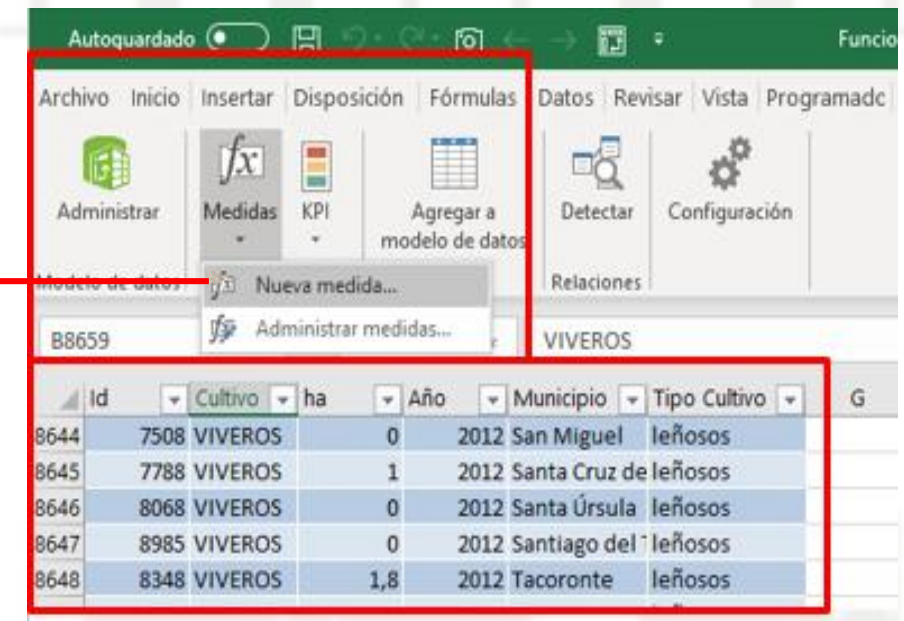
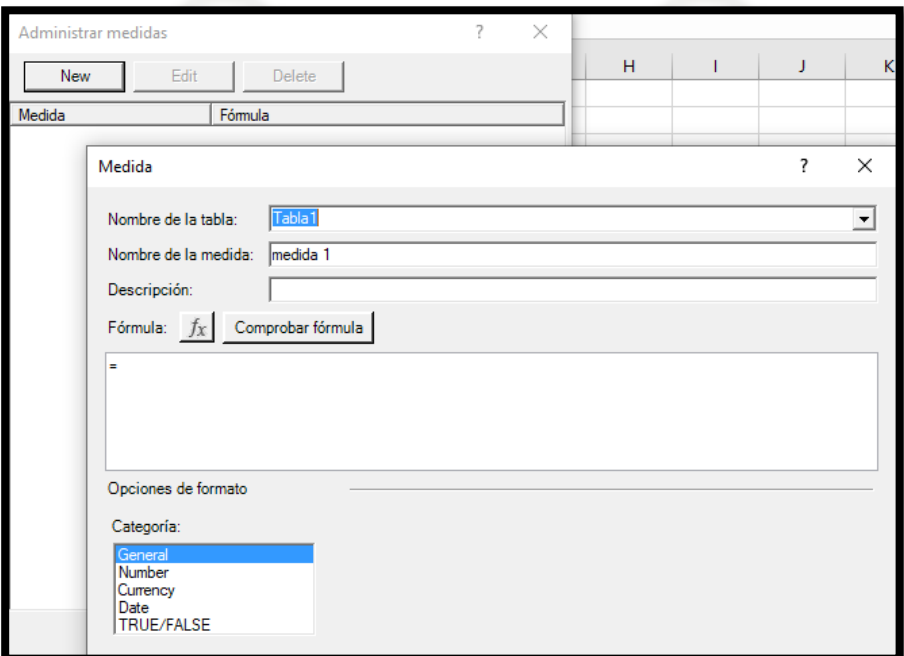
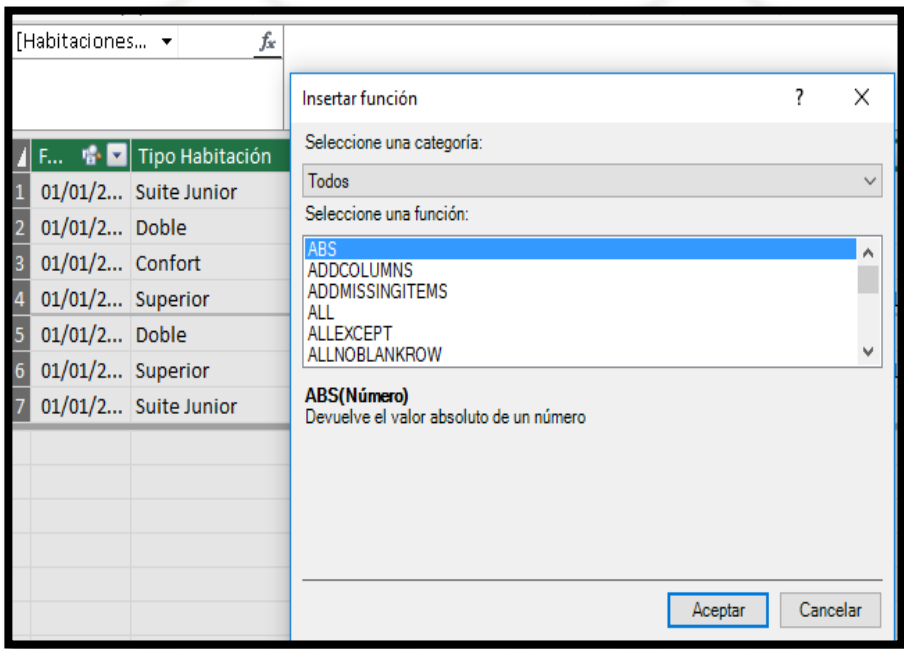
Si consideramos cambiar el nombre de la medida lo podemos hacer directamente en el código o en la propia etiqueta y cambiarle el nombre.

La ventaja es que el código DAX ha sido generado automáticamente, sin escribirlo nosotros.

Creación de medidas rápidas en Power Pivot

Igualmente en Power Pivot contamos con la opción de crear medidas rápidas y podremos acceder a ellas por diferentes vías entre ellas a través del icono Fx en la barra de fórmulas DAX de la ventana de Power Pivot que nos da acceso al cuadro de dialogo para introducir las fórmulas en DAX y al cuadro de elección de las funciones disponibles agrupadas en categorías relacionadas en grupos afines según nuestra consideración.

En la pestaña Power Pivot contamos con la opción **Administrar Medidas** que nos permite, no solo crear sino además gestionar las medidas .



2.3 Consejos relacionados con la creación de medidas y columnas calculadas

No hay “Columnas Desnudas” en las medidas (sin función asignada)

Al hacer referencia a una columna en una fórmula de medida, siempre tiene que ser “envuelta” en una función. Una referencia “desnuda” a una columna producirá un error en una medida. Echemos un vistazo a un ejemplo:

[My New Measure] = Sales[Margin]... Al dar al botón comprobar fórmula nos da un error

El valor de la columna ‘Margin’ en la tabla ‘Sales’ no puede ser determinado es necesario contar con una función,.. no puede estar desnuda. Todo lo siguiente sería válido:

=SUM(Sales[Margin]) =AVERAGE(Sales[Margin]) =MIN(Sales[Margin])=MAX(Sales[Margin])

Hacer referencia a las columnas “desnudas” es lo que las fórmulas de columnas calculadas hacen. Las medidas son agregaciones, y no aceptan referencias de columnas desnudas. Recuerde, las referencias de columnas desnudas son correcto en columnas calculadas. Esta regla sólo se aplica a las medidas

Mejores Prácticas: Referenciar Columnas y Medidas de Manera Diferente

Cada vez que escribo una fórmula de medida, sigo estas dos técnicas:

1. Para referenciar una columna, incluyo el nombre de la tabla:

NombredTabla[NombredColumna]

1. Para referencia a una medida, omitimos el nombre de la tabla:

[NombredMedida]

Esto facilita que las fórmulas sean más legibles y permite diferenciar claramente si estamos haciendo referencia en una fórmula a una columna calculada o a una medida.

Además, hay muchas situaciones en las que la omisión del nombre de la tabla en una referencia de columna devolverá un error. Después de esta buena práctica evita ese problema también.

Mejores Prácticas: Asignar Medidas a las Tablas Correctas

Mejores Prácticas: Asignar Medidas a las Tablas Correctas

El cuadro “nombre de tabla” en el editor de medida controla qué tabla se asignará la medida en la lista de campos. La medida será “vinculada” con la tabla de ventas en la lista de campos.

Regla Simple: Asigno mis medidas a las tablas que contienen las columnas numéricas utilizadas en la fórmula. Esto no es más que una buena higiene para que nuestro modelo sea más fácil de entender más tarde.

Las 2 reglas básicas de las funciones DAX

Regla A: las funciones DAX se evalúan según las tablas de origen, NO la tabla dinámica

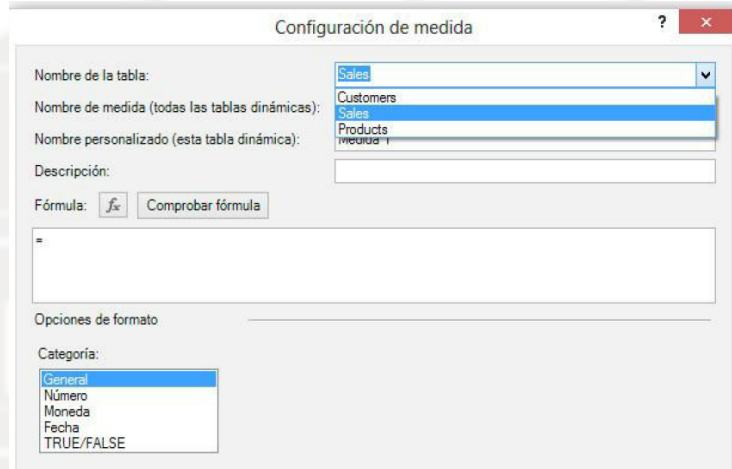
Es muy tentador pensar que la celda Total en la parte inferior de un TD es la suma de las celdas por encima de ella, pero esa no es la forma en que se calcula. En lo que se refiere a DAX, el hecho de que la celda de Gran Total coincida con la suma de los números anteriores roza en coincidencia.

Así que cuando estás pensando en cómo construir una fórmula de medida, o se depura una que no está funcionando bien, visualiza la tabla subyacente en la ventana de Power Pivot, porque el motor DAX está haciendo su trabajo en ese contexto.

Regla B: Cada Celda de la Medida es Calculada de manera Independiente

Al pensar en cómo se calcula tu medida, lo mejor es pensar en “una celda a la vez”. Por lo tanto, elige una celda y visualiza cómo ha sido calculada, como si se tratara de un elemento aislado e independiente.

El valor de una celda de medida NUNCA afecta el valor de otra celda medida. Las medidas se calculan de forma independiente, y se calculan en contra de la tabla(s) de origen.



ProductKey	Profit	Total Sales	Profit Pct
2001	\$287,087	\$652,367	44.0 %
312	\$236,794	\$547,475	43.3 %
328	\$2,831	\$6,292	45.0 %
344	\$47,462	\$98,600	48.1 %
2002	\$337,581	\$768,886	43.9 %
312	\$284,772	\$658,402	43.3 %
328	\$5,347	\$11,885	45.0 %
344	\$47,462	\$98,600	48.1 %
Total general	\$624,668	\$1,421,253	44.0 %

NO concuerdan

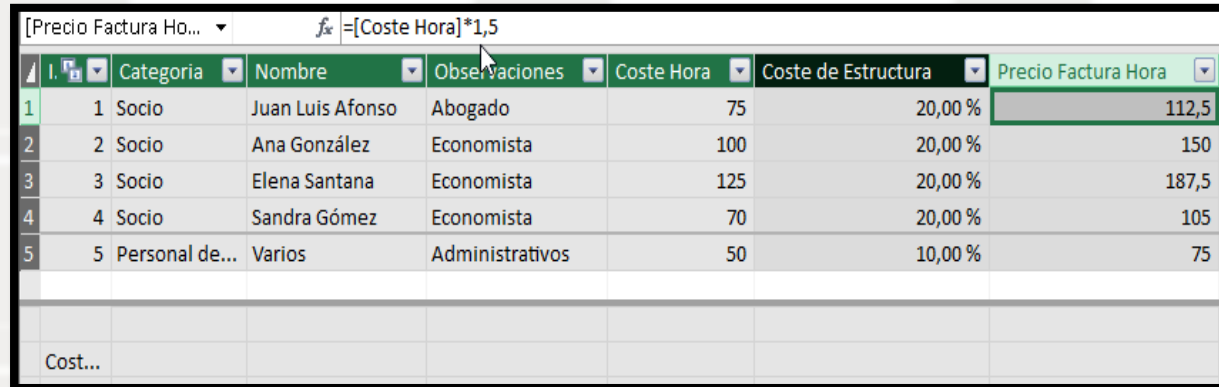
Promedio: 45.5 % Recuento: 6 Sum

Month	Profit
1	\$1,076,909
2	\$1,132,572
3	\$1,183,822
4	\$1,256,479
5	\$1,419,630
6	\$1,445,822
7	\$861,808
8	\$856,595
9	\$826,278
10	\$910,113
11	\$937,113
12	\$1,381,832
Grand Total	\$13,288,972

2.4 Aprendiendo a entender las funciones DAX

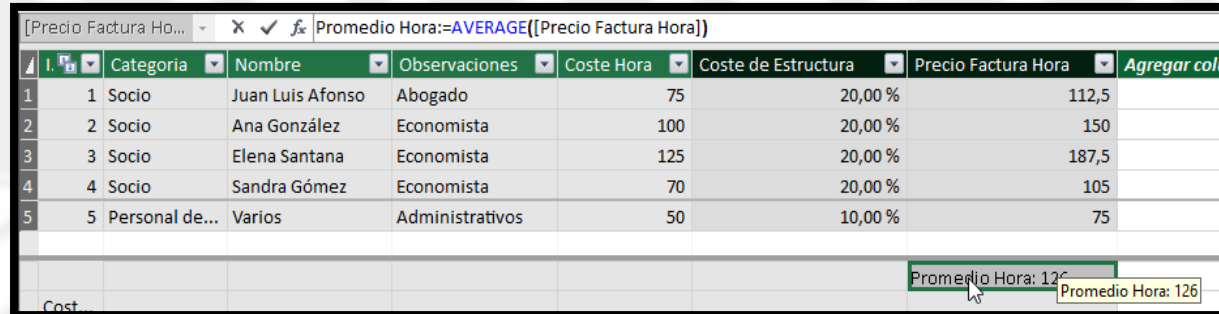
Sintaxis para columna calculada (=) y campo (medida) calculado (:=)

El operador de asignación de funciones es distinto para columnas calculadas que para campos calculados. Para columna calculada utiliza el signo igual (=), mientras que para las medidas se utiliza dos puntos e igual (:=) después del nombre y antes de escribir la fórmula.



The screenshot shows a table with the following columns: I., Categoría, Nombre, Observaciones, Coste Hora, Coste de Estructura, and Precio Factura Hora. The formula bar at the top displays the formula for the 'Precio Factura Hora' column: `=[Coste Hora]*1,5`. The table contains 5 rows of data.

I.	Categoría	Nombre	Observaciones	Coste Hora	Coste de Estructura	Precio Factura Hora
1	1 Socio	Juan Luis Afonso	Abogado	75	20,00 %	112,5
2	2 Socio	Ana González	Economista	100	20,00 %	150
3	3 Socio	Elena Santana	Economista	125	20,00 %	187,5
4	4 Socio	Sandra Gómez	Economista	70	20,00 %	105
5	5 Personal de...	Varios	Administrativos	50	10,00 %	75



The screenshot shows the same table as above, but with a calculated measure. The formula bar at the top displays the formula: `Promedio Hora:=AVERAGE([Precio Factura Hora])`. The table includes an additional column 'Agregar col...' and a summary row at the bottom showing the average value for the 'Precio Factura Hora' column.

I.	Categoría	Nombre	Observaciones	Coste Hora	Coste de Estructura	Precio Factura Hora	Agregar col...
1	1 Socio	Juan Luis Afonso	Abogado	75	20,00 %	112,5	
2	2 Socio	Ana González	Economista	100	20,00 %	150	
3	3 Socio	Elena Santana	Economista	125	20,00 %	187,5	
4	4 Socio	Sandra Gómez	Economista	70	20,00 %	105	
5	5 Personal de...	Varios	Administrativos	50	10,00 %	75	
						Promedio Hora: 126	Promedio Hora: 126

Referenciando a una tabla o una columna

En caso de tener que referenciar en una fórmula a una tabla y/o columna de nuestro modelo de datos el procedimiento a seguir es el siguiente:

Para escribir un nombre de una tabla. *Empiece a escribir el nombre de la tabla.* La función Autocompletar fórmula proporciona una lista desplegable que contiene nombres válidos que comienzan con esas letras.

Para escribir el nombre de una columna. *Escriba un paréntesis y elija la columna en la lista de columnas de la tabla actual.* Para una columna de otra tabla, empiece a escribir las primeras letras del nombre de la tabla y, a continuación, elija la columna en la lista desplegable Autocompletar.

Usar varias funciones en una fórmula

Las funciones se pueden anidar, es decir, puede usar los resultados de una función como argumento de otra función. Muchas funciones de DAX están diseñadas para usarse exclusivamente como funciones anidadas. Estas funciones devuelven una tabla, que no se puede guardar directamente como resultado en el libro de PowerPivot, pero que se debe proporcionar como entrada para una función de tabla. Por ejemplo, las funciones SUMX, AVERAGEX y MINX requieren una tabla como primer argumento.

Tipos de operaciones y de datos. Aritméticos, comparación, lógicos, texto, blob, etc

El tipo de operaciones que podemos realizar en nuestro modelo de datos son muy similares a las disponibles en Excel y que pasamos a presentar

Operadores Aritméticos

Suma	+
Sustracción	-
Multiplicación	*
División	/
Potenciación	^

Operadores de comparación

Igualdad	=
Diferente de	<>
Mayor Que	>
Mayor o Igual Que	>=
Menor Que	<
Menor o Igual Que	<=

Operadores lógicos

Conjunción (Y)	&&	Similar a AND
Negación	!	Similar a NOT
Disyunción Inclusiva (O)		Similar a OR

Operadores de texto

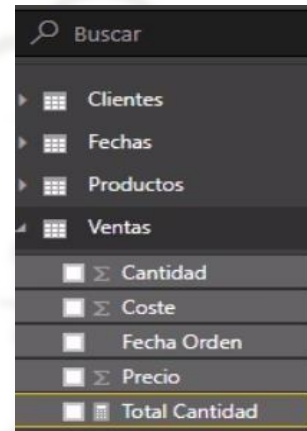
Concatenación &

2.5 Entendiendo lo que son los iteradores. Funciones Sufijo X

Supongamos que contamos con la tabla de ventas con los campos mostrados, si quisiéramos calcular el total de ventas tendríamos que multiplicar para cada fila el campo cantidad por el campo precio y su suma nos daría el total de ventas. En DAX esta operación no funcionaría:

SUM(Ventas [Cantidad]*Ventas[Precio]) No se puede hacer

La función SUM suma todos los números de una columna y solo admite una sola columna, para obtener el resultado correcto debemos usar la función SUMX que devuelve la suma de una expresión evaluada para cada fila de una tabla, esta función es un iterador en DAX. Esta función espera dos parámetros una tabla y una expresión



```
Total Ventas = sumx
```

SUMX(Tabla; Expresión)
Devuelve la suma de una expresión evaluada para cada fila de una tabla.

SUMX Devuelve la suma de una expresión evaluada para cada fila de una tabla.

Así hemos creado una medida con la función SUMX que recorre la tabla fila a fila y para cada fila calcula la multiplicación de cantidad por precio, este resultado por fila se almacena en memoria y cuando SUMX termina de recorrer o iterar toda la tabla suma el resultado obtenido por filas

```
Total Ventas = sumx(Ventas; Ventas[Cantidad] * Ventas[Precio])
```

De la misma forma calculamos el total de beneficio, *Medida Total Beneficio*

```
Total Beneficio = sumx(Ventas; Ventas[Cantidad] * (Ventas[Precio] - Ventas[Coste]))
```

Finalmente, para obtener el % de beneficio creamos una nueva medida que llamaremos % de beneficio y para el cálculo del cociente entre la medida Total Beneficio / Total Ventas utilizaremos la función DIVIDE que nos permitiría controlar las divisiones por cero y además la acción ante este tipo de error (AlternateResult) que si no lo indicamos nos daría un valor vacío.

```
% Beneficio = DIVIDE([Total Beneficio]; Ventas[Total Ventas];  
DIVIDE(Numerador; Denominador; [AlternateResult])  
Función de división segura con posibilidad de controlar divisiones por  
cero.
```

Destacar que estamos utilizando las medidas generadas anteriormente y que las medidas no necesitan utilizar el nombre de tablas asociada, es opcional, en las columnas. Por último, ajustamos el formato de salida a % tal y como se muestra en el Grupo Formato. Finalmente, para ver el resultado de este cálculo cuando se utiliza en un informe, vamos a escoger desde el panel de visualización un objeto tipo tabla

The screenshot shows the Power BI interface with a table of sales data. The table has four columns: 'Categoría', 'Total Ventas', 'Total Beneficio', and '% Beneficio'. The data is as follows:

Categoría	Total Ventas	Total Beneficio	% Beneficio
Accesorios	700.759,96	438.674,57	62,60 %
Bicicletas	28.318.144,65	11.505.796,50	40,63 %
Ropa	339.772,61	136.412,58	40,15 %
Total	29.358.677,22	12.080.883,65	41,15 %

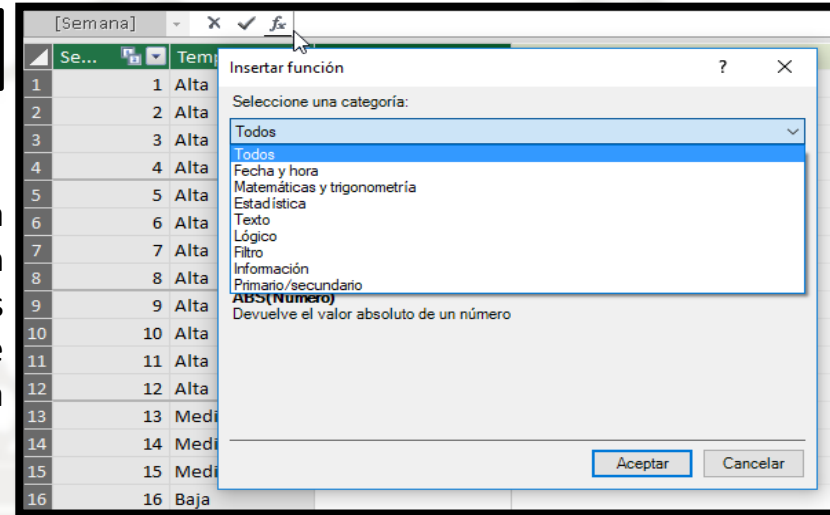
On the right side, the 'CAMPOS' panel is visible, showing a list of fields. The '% Beneficio' field is highlighted with a yellow border. Below the table, there are sections for 'VALORES' (Values) and 'FILTROS' (Filters). The 'VALORES' section lists 'Categoría', 'Total Ventas', 'Total Beneficio', and '% Beneficio'. The 'FILTROS' section shows 'Filtros de nivel de objeto vis...' and lists '% Beneficio (Todos)', 'Categoría (Todos)', and 'Total Beneficio (Todos)'. The 'CAMPOS' panel also shows a search bar and a list of fields including 'Color', 'Cantidad', 'Coste', 'Fecha Orden', 'Precio', 'Total Beneficio', 'Total Cantidad', and 'Total Ventas'.

This is a close-up of the 'CAMPOS' panel in Power BI. It features a search bar at the top with the text 'Buscar'. Below the search bar, there is a list of fields. The field '% Beneficio' is highlighted with a yellow border. Other fields in the list include 'Color', 'Cantidad', 'Coste', 'Fecha Orden', 'Precio', 'Total Beneficio', 'Total Cantidad', and 'Total Ventas'. The panel also shows a search bar and a list of fields including 'Color', 'Cantidad', 'Coste', 'Fecha Orden', 'Precio', 'Total Beneficio', 'Total Cantidad', and 'Total Ventas'.

3. Principales funciones DAX

3.1 Introducción

Como expusimos con anterioridad a través del icono f_x en la barra de fórmulas DAX accedemos al cuadro de dialogo para la elección de las funciones disponibles agrupadas en categorías y que hemos relacionado en grupos afines según nuestra consideración y que pasamos a analizar a continuación las mas relevantes para familiarizarnos con el lenguaje de funciones DAX



- a) Funciones de Agregación
- b) Funciones de Conteo
- c) Funciones Lógicas
- d) Funciones de Fecha y Hora
- e) Funciones de Información
- f) Funciones de Texto

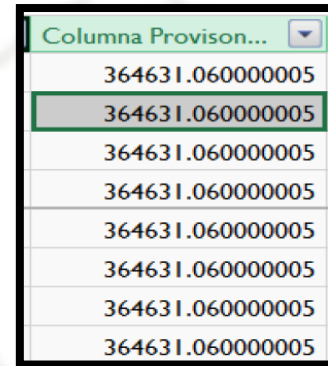


1. Fecha y Hora
2. Matemáticas y Trigonométricas
3. Estadísticas
4. Texto
5. Lógicas
6. Filtro
7. Información
8. Primario / Secundario
- 9. Time Intelligence**
- 10. Misceláneas**

3.2 Funciones de agregación: SUM, MAX, PRODUCT, COUNT, etc

Las agregaciones permiten resumir o agrupar datos. Se encargan de realizar un cálculo sobre una columna a partir del máximo nivel de detalle de sus filas, afectando en principio a todas ellas, salvo que indiquemos algún tipo de filtro o desglose descriptivo. Así, las funciones de agregación siempre se aplican a la columna entera y se expanden si se agregan más registros.

Para entender esta categoría de funciones vamos a crear la siguiente columna calculada de forma provisional, =SUM(TABLA_Pedidos[Costo Total]). Observemos el resultado. Todos los valores de las distintas celdas de la columna hallados con la función de agregación SUMA son iguales



Columna Provision...
364631.060000005
364631.060000005
364631.060000005
364631.060000005
364631.060000005
364631.060000005
364631.060000005
364631.060000005

Las Funciones de Agregación disponibles son:

SUM	PRODUCT	AVERAGE	STDEV.S	STDEV.P
MAX	MIN	VAR.P	VAR.S	MEDIAN
PERCENTILE.EXC	PERCENTILE.INC	DISTINCCOUNT	COUNT	

Estas funciones las podemos clasificar en 4 tipos:

1. **Aditivas:** Agregan sin restricciones.

Funciones: SUM, COUNT, COUNTROWS, AVERAGE

2. **Semi-Aditivas:** Agregan sólo en parte, veamos un ejemplo: necesitamos saber el total de clientes distintos que compran en nuestras tiendas, sabemos que en enero han sido 200 clientes, en febrero han sido 225 clientes y en marzo 210; en cambio si lo que queremos saber el total de clientes distintos que compran en nuestras tiendas en el primer trimestre, no podemos sumar tal cual, porque habrá clientes que nos hayan comprado durante los tres meses de este trimestre y otros que no, por tanto, será un número comprendido entre 225 y 635.

Funciones: DISTINCTCOUNT

3. **No Aditivas:** No pueden agregar.

Funciones: MIN, MAX

4. **Personalizadas:** decidimos nosotros mediante diversas expresiones DAX cómo deben actuar. Por ejemplo, si queremos calcular un balance de situación, habrá cuentas que sumen y cuentas que resten.

Funciones: no hay ninguna específica, tendremos que crear expresiones DAX más complejas que resuelvan de forma individual cada necesidad.

En la siguiente tabla mostramos la sintaxis y una breve descripción

SUM(<Columna>)	Suma todas las filas de una columna. = SUM([ImporteVenta])
COUNT(<Columna>)	Cuenta el número de filas de una columna que tenga datos numéricos o fechas. = COUNT([ImporteVenta])
COUNTROWS(<Tabla>)	Cuenta el número de filas de una tabla. = COUNTROWS('Cliente') = COUNTROWS(RELATEDTABLE('Ventas'))
AVERAGE(<Columna>)	Devuelve el promedio de todos los números de una columna. = AVERAGE([ImporteVenta])
DISTINCTCOUNT(<Columna>)	Cuenta el número de filas con valores distintos de una columna. = DISTINCTCOUNT(Ventas[Ticket])
MIN(<Columna>)	Devuelve el menor valor numérico de una columna que tenga datos numéricos o fechas. = MIN(Ventas[TiendaSK])
MAX(<Columna>)	Devuelve el mayor valor numérico de una columna que tenga datos numéricos o fechas. = MAX(Ventas[TiendaSK])

Ejemplo

Queremos calcular el total de ventas de la empresa, crearemos una columna calculada utilizando la función SUM(): Esta función vista nos ofrece la suma de todos los valores de la columna “[ImporteVenta]” y le aplicamos el formato adecuado. Si creamos una tabla dinámica arrastramos “TotalVentas” al área de valores, nos muestra el resultado de sumar “[ImporteVenta]” para todas las filas de la tabla y ya formateado.

Ahora lo desglosamos por Tienda y una columna por cada año:

=SUM(Ventas[ImporteVenta])		
Cantidad	ImporteVenta	
1		0,74
1		0,74
1		0,74
1		0,74
1		0,74
		TotalVentas: 1.078.749,83

TotalVentas	Etiquetas de columna		
Etiquetas de fila	2012	2013	Total general
Tienda Alicante 01	322.513,71	86.304,34	408.818,05
Tienda Alicante 02	42.516,38	11.720,06	54.236,44
Tienda Castelldefels	244.255,51	76.325,37	320.580,88
Tienda La Coruña	75.213,03	28.051,11	103.264,14
Tienda Murcia 01	36.336,70	12.288,01	48.624,71
Tienda Principal	113.390,62	29.834,99	143.225,61
Total general	834.225,95	244.523,88	1.078.749,83

Arrastrar campos entre las áreas siguientes:

FILTROS	COLUMNAS
	Año
FILAS	VALORES
Tienda	TotalVentas

3.3 Funciones de Conteo

Estas funciones tienen como objetivo contar todos los elementos de una columna que cumplan una serie de condiciones. Entre las más relevantes destacamos las siguientes:

COUNT	Cuenta el número de celdas en una columna que tienen números.
COUNTA	Cuenta el número de celdas que tienen números, valores lógicos y texto en columna.
COUNTBLANK	Cuenta el número de celdas vacías en una columna.
COUNTROWS	Cuenta el número de filas que tiene una tabla.
DISTINCTCOUNT	Cuenta el número de celdas diferentes en una columna.

Ejemplo

id	Fecha	Cliente	Activ...	Horas	Personal	
1	15/01/2016 0:00:00		6	7	150	5
2	22/02/2016 0:00:00			7	20	5
3	03/02/2016 0:00:00		0	7	160	5
4	04/04/2016 0:00:00		8	7	90	5

1 Count: 33
2 CountA: 33
3 Countblank: 1
4 Countrows: 33
5 Distinctcount: 20

- 1 Count:=**Count**(Tabla_D_partes[Horas])
- 2 CountA:=**COUNTA**(Tabla_D_partes[Horas])
- 3 Countblank:=**COUNTBLANK**(Tabla_D_partes[Cliente])
- 4 Countrows:=**COUNTROWS**(Tabla_D_partes)
- 5 Distinctcount:=**DISTINCTCOUNT**(Tabla_D_partes[Horas])

3.4 Funciones Lógicas

Estas funciones son muy similares a las disponibles en Excel. Presentamos a continuación las más relevantes y su equivalente en Excel

<i>Función Excel</i>	<i>Función -DAX</i>	<i>Descripción</i>
Y	AND	Comprueba si todos los argumentos son VERDADEROS y devuelve VERDADERO o FALSO. Devuelve FALSO si alguno de los argumentos es FALSO
Falso	FALSE	Devuelve el valor lógico FALSO
SI	IF	Comprueba si se cumple una condición y devuelve un valor si se evalúa como VERDADERO y otro valor si se evalúa como FALSO.
SI.ERROR	IFERROR	Retorna lo especificado en el segundo argumento si hay un error. Devuelve un valor si la expresión es un error y otro valor si no lo es
NO	NOT	Cambia FALSO por VERDADERO y VERDADERO por FALSO.
O	OR	Comprueba si alguno de los argumentos es VERDADERO y devuelve VERDADERO o FALSO. Devuelve FALSO si todos los argumentos son FALSO
Verdadero	TRUE	Devuelve el valor lógico VERDADERO

3.5 Funciones fecha y hora sin Time Intelligence

Igualmente, este grupo de funciones son similares a las de Excel y su sintaxis también. Pasamos a presentar alguna de ellas. Y su equivalente en Excel

Función Excel	Función DAX	Descripción
Fecha	DATE	Devuelve el número que representa la fecha en código de fecha y hora
FechaNumero	DATEVALUE	Convierte una fecha en forma de texto en un número que representa la fecha en código de fecha y hora
HORA	HOUR	Devuelve la hora como un número de 0 (12:00 a.m.) a 23 (11:00 p.m.)
Hoy	TODAY	Devuelve la fecha actual
DIASEM	WEEKDAY	Devuelve un número de 1 a 7 que identifica el día de la semana.

Ejemplo

	Fecha	Dia de la semana	Cliente
1	15/01/2016 0:00:00	6	6
2	22/02/2016 0:00:00	2	
3	03/02/2016 0:00:00	4	0
4	04/04/2016 0:00:00	2	8

3.6 Funciones de información

Cada vez que necesitamos verificar o analizar el tipo de dato de una columna, las funciones de información vienen al rescate, al igual que la gran mayoría de las anteriores, son muy similares a las de Excel.

Todas las funciones de información devuelven uno de dos valores. FALSE (Falso) o TRUE (Verdadero). Rescatamos las más intuitivas en la tabla enseguida.

Función	Descripción
ISBLANK	Verifica si el valor es BLANK() devolviendo TRUE si es afirmativo, de otro modo FALSE.
ISERROR	Verifica si el valor es un Error devolviendo TRUE si es afirmativo, de otro modo FALSE.
ISEVEN	Verifica si el valor numérico es par, devolviendo TRUE si es afirmativo, de otro modo FALSE.
ISLOGICAL	Verifica si el valor es Booleano devolviendo TRUE si es afirmativo, de otro modo FALSE.
ISNOTEXT	Verifica si el valor NO es tipo texto devolviendo TRUE si es afirmativo, de otro modo FALSE.
ISNUMBER	Verifica si el valor es de tipo numérico (Integer, Double, Currency) devolviendo TRUE si es afirmativo, de otro modo FALSE.
ISTEXT	Verifica si el valor es de tipo texto devolviendo TRUE si es afirmativo, de otro modo FALSE.

3.7 Funciones de Texto

Casi todas las funciones de texto DAX son similares a las que están en Excel, con algunas pocas excepciones.

Función	Descripción
CONCATENATE	Une dos cadenas de texto.
EXACT	Compara dos cadenas de texto, devuelve True si son idénticas de otro modo devuelve False.
FIND	Devuelve la posición de una cadena de texto dentro de otra.
FIXED	Redondea un número a las posiciones decimales especificadas como tipo texto.
FORMAT	Convierte un valor de acuerdo a un formato señalado.
LEFT	Retorna el número de caracteres especificados de una cadena de texto empezando desde la izquierda.
LEN	Devuelve el número de caracteres en una cadena de texto.
LOWER	Convierte todas las letras a minúsculas
MID	Devuelve el número de caracteres en una cadena de texto empezando desde una posición indicada.
REPLACE	Reemplaza parte de una cadena de texto basado en un número de caracteres especificados.
REPT	Repite un carácter dado un número de veces señalado.
SEARCH	Devuelve el número de posición de un carácter
SUBSTITUTE	Reemplaza un texto especificado por uno nuevo.
TRIM	Remueve todos los espacios en una cadena de texto exceptuando los espacios únicos que hay entre palabras.
UPPER	Convierte todas las letras a Mayúsculas.
VALUE	Convierte una cadena de texto que representa un número a tipo numérico.

3.8 Funciones de navegación entre tablas. RELATED y RELATEDTABLE

Estas funciones nos permiten navegar y obtener valores de columnas de diversas tablas del modelo siempre que existan relaciones entre ellas.

En la siguiente tabla mostramos la sintaxis de estas funciones y una breve descripción. Para conocer más detalles puede ver la ayuda del producto.

RELATED(<Columna>)	Devuelve un valor relacionado de la columna de otra tabla siguiendo la relación M->1. Ejemplo: agregar una columna calculada con la “[Poblacion]” a la tabla “Tienda” (Muchas Tiendas -> 1 Población): = RELATED(Geografia[Poblacion])
RELATEDTABLE(<Tabla>)	Devuelve una tabla en un contexto especificado por los filtros indicados, siguiendo la relación 1->M. Ejemplo: agregar una columna con el número de clientes a la tabla “Geografia”: = COUNTROWS (RELATEDTABLE ('Cliente'))

La función RELATED se puede utilizar, por ejemplo, para desnormalizar, creando columnas calculadas que permitan añadir columnas de varias tablas del origen, que estén relacionadas, a una sola tabla.

Sobre la función RELATEDTABLE se utiliza en conjunto con otras funciones

4. Consejos y consideraciones a tener en cuenta en las formulas DAX

4.1 Evitar obtener un error como resultado de una formula

Las formulas DAX pueden arrojar como resultado un error derivado de diversas causas, muy similares a las fórmulas de Excel, como son: dividir por cero, operar con un tipo de valor no correcto o del tipo requerido como sumar un número y texto, etc

A lo largo de los años se han distinguido tres categorías de errores, estas son:

- **Errores del tipo requerido**, estos se producen generalmente en el proceso de importación de los datos a nuestro modelo donde estos se convierten automáticamente y puede dar lugar a, por ejemplo, convertir un número en texto. Por tanto, es necesario una vez importados los datos a nuestro modelo verificar la integridad y el tipo requerido de los mismos sea el correcto.
- **Errores matemáticos** derivados de significado e incoherencia matemática del resultado como sería, dividir entre cero en cuyo caso el resultado sería ∞ (infinito) al contrario que las fórmulas Excel que en estos casos da el error #¡DIV/0!
- **Errores faltan datos o hay vacíos**. Este error es similar al #N/A de Excel, es decir, nos indica que no está disponible el valor deseado y que la fórmula no podrá mostrar el resultado correcto. Sin embargo, a diferencia de Excel es que el DAX no lo trata como un error y por lo tanto afecta a toda la columna, es un tipo de valor especial que lo identifica con BLANK.

NOTA

¡OJO! Si una fila resulta en error, entonces todos los demás valores de la columna serán errores independientemente si la operación es válida para filas específicas, el hecho de encontrar un solo error propaga este mismo en toda la columna, por esto es necesario manejarlos e interceptarlos correctamente

Evitar errores de conversión

Para evitar errores de conversión es una buena práctica asegurarse que el tipo de datos que esperamos de una columna sea el que necesitamos, para ello contamos con las funciones de conversión y la función lógica IFERROR, por ejemplo:

[Nueva Columna]=IFERROR(VALUE([Precio]) *VALUE([Unidades]); BLANK())

4.2 Errores Matemáticos

Estos errores se producen porque matemáticamente son desconocidos o carecen de significado, ellos son:

- División entre cero
- Raíz Cuadrada de un número negativo
- Logaritmo de un número negativo

División por cero

La primera situación, división por cero, es bastante especial porque en lugar de devolver un error, retorna un valor especial: infinito (∞). Esto varía respecto a Excel quien devuelve #¡DIV/0! Cuando una división por cero se ha realizado. Existe una función llamada DIVIDE que nos permite realizar una división y a su vez definir qué valor queremos retornar cuando encuentre división entre 0.

Valores vacíos

Los valores vacíos son datos faltantes o desconocidos en la base de datos. El lenguaje DAX no los trata como un error por lo que no daña toda la columna si llegan a aparecer, para el lenguaje DAX es un tipo especial de valor, algo similar a lo que sucede con el tipo especial NaN.

Para el modelo de datos una celda vacía, valor faltante o valor en blanco es lo mismo: lo identifica con BLANK, el cual es una forma de representar esta condición especial de ausencia de valor.

Contamos con una función BLANK() para retornar este tipo especial de valor, el cual es diferente de una cadena de caracteres vacía (“”) ya que cada uno tiene comportamientos diferentes.

Destacar

Cada vez que desee devolver un valor vacío utilice la función BLANK(), de ningún modo utilice expresiones como: “” o “ ”. BLANK() es nuestro amigo.

4.3 Claridad, Reglas para la presentación de fórmulas DAX óptima

Cuando creamos una fórmula DAX larga es una buena práctica acostumbrarnos a darle una forma que sea legible, no solamente para un tercero, sino también para nosotros mismos, ya que después de un tiempo de haberla desarrollado y necesitemos ajustarla, modificarla o reciclarla, sea sencillo y rápido. A continuación, se presentan una serie de pautas para alcanzar este objetivo.

- Siempre incluir un espacio después de un paréntesis.
- Poner un espacio antes y después de un operador sin importar su categoría.
- Si se tiene que dividir una expresión en múltiples líneas, poner el operador al principio de la nueva línea.
- Una función que se ha dividido en múltiples líneas debe siempre tener sus distintos argumentos en líneas diferentes.
- Omitir espacio entre el nombre de la tabla y el nombre de la columna.
- Utilizar las comillas simples cuando sea estrictamente necesario, es decir, cuando el nombre de una tabla contenga espacios.
- Omitir el nombre de la tabla cuando se está haciendo referencia a una medida.
- Poner un espacio después de un argumento si estos están en la misma línea. (Shift + Enter)
- Siempre utilizar el nombre de las tablas cuando se haga referencia a una columna
- Poner una función en la misma línea solamente si tiene un único argumento.
- Si una función está escrita en más de una línea, utilice Tab para indexar, así mismo, los paréntesis de apertura y cierre deben estar alineados.

5 Ejemplos de empleo de funciones DAX frecuentes

Una vez sentadas las bases, estamos capacitados para ir aprendiendo cualquier función DAX e ir aplicándola. Vamos a conocer algunas de uso frecuente, que además suelen utilizar en combinación con otras existentes

5.1 Uso de la función COUNTROWS y con filtro

La función COUNTROWS nos retorna un escalar, por ello de agregación, su único argumento acepta una tabla. Como su nombre indica cuenta el número de filas en una tabla o, en una tabla definida por una expresión.

Sintaxis: COUNTROWS(<tabla>)

Según la definición podemos utilizar COUNTROWS así:

= COUNTROWS (TABLA_Pedidos)

Lo cual cuenta el número de filas en la tabla Pedidos, sin embargo, también nos dice que en dicho argumento podemos utilizar expresiones que devuelva una tabla, por lo que podemos crear la siguiente medida:

[Ventas de Alta Rentabilidad]:=COUNTROWS (FILTER (TABLA_Pedidos; TABLA_Pedidos[Precio de Venta] >= 40))

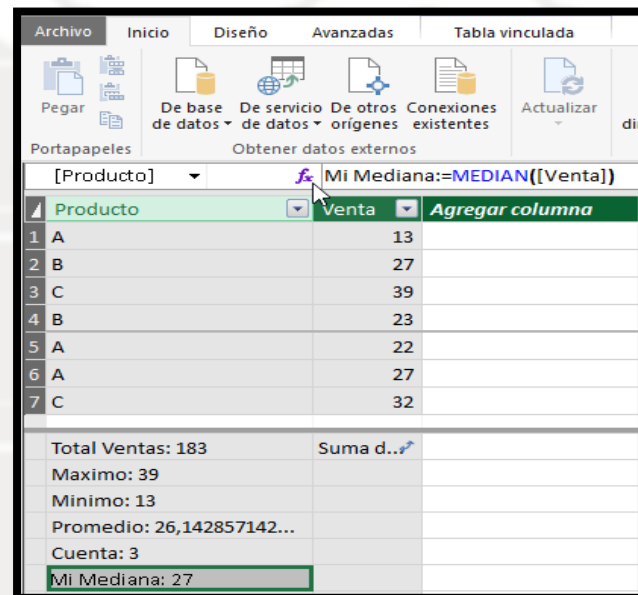
En la expresión DAX, la función FILTER retorna una tabla con solamente aquellas filas que en la columna Precio de Venta tienen un valor mayor o igual a 40 y, como esta encapsulada en la función COUNTROWS, entonces la medida retorna el número de ventas con un precio mayor o igual a 40 para el contexto actual.

5.2 Uso de la función Mediana

En este caso queremos calcular la Mediana de las ventas de nuestros productos, función que no existe y para ello vamos a crear una medida o campo calculado utilizando el lenguaje DAX y en concreto la función MEDIAN, tal y como vemos en la siguiente ilustración

Así en el área de medidas aparece la recién creada, junto con el resultado aplicado a toda la tabla.

Si ahora creamos una tabla dinámica basada en nuestro modelo de datos podemos observar como la medida recién creada la tendremos disponible para incorporar a nuestra tabla



The screenshot shows the Power BI Desktop interface. At the top, there are tabs for 'Archivo', 'Inicio', 'Diseño', 'Avanzadas', and 'Tabla vinculada'. Below the tabs is a ribbon with various icons for actions like 'Pegar', 'De base de datos', 'De servicio de datos', 'De otros orígenes', 'Conexiones existentes', and 'Actualizar'. Below the ribbon, there is a 'Portapapeles' section and an 'Obtener datos externos' section. The main area shows a table with columns 'Producto' and 'Venta'. The table has 7 rows of data. Below the table, there is a summary section with various statistics: 'Total Ventas: 183', 'Maximo: 39', 'Minimo: 13', 'Promedio: 26,142857142...', 'Cuenta: 3', and 'Mi Mediana: 27'. The 'Mi Mediana: 27' row is highlighted with a green border.

Producto	Venta
1 A	13
2 B	27
3 C	39
4 B	23
5 A	22
6 A	27
7 C	32

Total Ventas: 183	Suma d...
Maximo: 39	
Minimo: 13	
Promedio: 26,142857142...	
Cuenta: 3	
Mi Mediana: 27	

5.3 Uso de las funciones COUNTROWS() y DISTINCTCOUNT().

Calcular el promedio de ventas y las ventas medias diarias

Nuestro Jefe de Dpto nos solicita un informe de ventas por día y ventas por transacciones, agrupadas por mes. ¿Cómo construir ese informe en Excel normal? No podías escribir fórmulas en un TD. Habría que hacer algo de alquimia de fórmula muy seria para conseguir que funcione. Si además otro jefe nos solicitara lo mismo pero incorporando un segmentación por productos el problema comienza a complicarse, cambiar el informe normal de Excel a ser agrupado por producto en lugar de meses (y se puede rebanar por Año) no sería una modificación. Eso sería partir de cero, en muchos sentidos, y la reconstrucción de todo el informe. Con Power Pivot, sólo tienes que arrastrar los campos a la lista de campos.

Para ello vamos a ver la potencia de las medidas usando en este caso las funciones COUNTROWS() y DISTINCTCOUNT() para lo cual vamos a calcular dos medidas:

[Transactions] = COUNTROWS(Sales) y

[Days Selling] = DISTINCTCOUNT(Sales[OrderDate])

Y poniendo estas medidas en nuestra TD el resultado es el que se adjunta

COUNTROWS(Sales), esta función devuelve el número de filas de la tabla especificada. Así, por ejemplo, en la figura anterior, hay 5017 filas en la tabla de ventas que tienen en el mes numero 1. A esta medida el hemos llamado [Transacciones] porque sé que cada fila en la tabla de ventas es una transacción. Pero si una sola transacción se divide en varias filas, no podía hacer eso.

Row Labels	Total Sales	Days Selling	Transactions
1	\$2,375,857	93	5,017
2	\$2,502,387	85	5,059
3	\$2,610,615	93	5,178
4	\$2,778,842	90	5,589
5	\$3,114,646	93	6,064
6	\$3,180,924	90	6,080
7	\$1,911,263	124	4,019
8	\$1,899,607	92	4,256
9	\$1,834,668	89	4,229
10	\$2,009,169	92	4,536
11	\$2,076,070	90	4,536
12	\$3,064,630	93	5,835
Grand Total	\$29,358,677	1124	60,398

Tendría que utilizar DISTINCTCOUNT() en una columna ID de transacción, que no tengo en el ejemplo DISTINCTCOUNT(Sales[OrderDate]) esta función devuelve de la tabla Sales y del campo OrderDate el número de valores distintos (únicos) de la columna que especifiques. Así, mientras que hay 5,017 filas para MonthNum 1, y todos ellos, obviamente, tienen un valor para el [OrderDate] columna, sólo hay 93 diferentes valores únicos para [OrderDate] en las 5017 filas.

Creación de dos nuevas medidas dependientes de las anteriores.

Ahora defino dos medidas más que dependen de las dos medidas anteriores.

$$[\text{Sales per Transaction}] = [\text{Total Sales}] / [\text{Transactions}] \text{ y}$$

$$[\text{Sales per Day}] = [\text{Total Sales}] / [\text{Days Selling}]$$

Resultados:

Row Labels	Total Sales	Days Selling	Transactions	Sales per Transaction	Sales per Day
1	\$2,375,857	93	5,017	\$473.56	\$25,547
2	\$2,502,387	85	5,059	\$494.64	\$29,440
3	\$2,610,615	93	5,178	\$504.17	\$28,071
4	\$2,778,842	90	5,589	\$497.20	\$30,876
5	\$3,114,646	93	6,064	\$513.63	\$33,491
6	\$3,180,924	90	6,080	\$523.18	\$35,344
7	\$1,911,263	124	4,019	\$475.56	\$15,413
8	\$1,899,607	92	4,256	\$446.34	\$20,648
9	\$1,834,668	89	4,229	\$433.83	\$20,614
10	\$2,009,169	92	4,536	\$442.94	\$21,839
11	\$2,076,070	90	4,536	\$457.69	\$23,067
12	\$3,064,630	93	5,835	\$525.22	\$32,953
Grand Total	\$29,358,677	1124	60,392	\$486.00	\$26,120

