

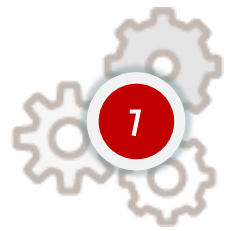
# Énfasis en Contexto de Evaluación

Cálculos DAX y Contexto de Evaluación

Experto en Lenguaje DAX



# Columnas Calculadas



# Introducción al Lenguaje DAX

Los Cuatro Sabores de Cálculos DAX + Dos Postres y un Aroma

## INT

INT retorna el número dado redondeado siempre hacia abajo al entero más cercano como tipo de dato INTEGER.

INT ( < Número > )

 [Más Información Aquí](#)



## Columnas Calculadas



Días de Entrega =

**INT** ( Pedidos[Fecha de Llegada] - Pedidos[Fecha de Pedido] )

País	Fecha de Pedido	Fecha de Envío	Fecha de Llegada	Días de Entrega
Colombia	41005	07/04/2012	41020	15
Argentina	07/04/2012	08/04/2012	20/04/2012	
Argentina	07/04/2012	07/04/2012	22/04/2012	
Colombia	08/04/2012	09/04/2012	04/05/2012	

# Introducción al Lenguaje DAX

Los Cuatro Sabores de Cálculos DAX + Dos Postres y un Aroma

## INT

INT retorna el número dado redondeado siempre hacia abajo al entero más cercano como tipo de dato INTEGER.

INT ( < Número > )

[Más Información Aquí](#)



## Columnas Calculadas



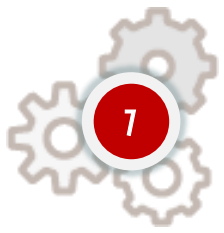
Días de Entrega =

**INT** ( Pedidos[Fecha de Llegada] - Pedidos[Fecha de Pedido] )

País	Fecha de Pedido	Fecha de Envío	Fecha de Llegada	Días de Entrega
Colombia	41005	07/04/2012	41020	15
Argentina	07/04/2012	08/04/2012	20/04/2012	
Argentina	07/04/2012	07/04/2012	22/04/2012	
Colombia	08/04/2012	09/04/2012	04/05/2012	

# Introducción al Lenguaje DAX

Los Cuatro Sabores de Cálculos DAX + Dos Postres y un Aroma



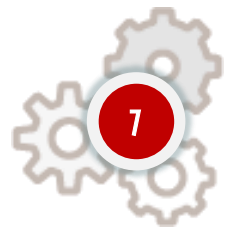
## Columnas Calculadas



Días de Entrega =

**INT** ( Pedidos[Fecha de Llegada] - Pedidos[Fecha de Pedido] )

País	Fecha de Pedido	Fecha de Envío	Fecha de Llegada	Días de Entrega
Colombia	06/04/2012	07/04/2012	21/04/2012	15
Argentina	41006	08/04/2012	41019	13
Argentina	07/04/2012	07/04/2012	22/04/2012	
Colombia	08/04/2012	09/04/2012	04/05/2012	



# Introducción al Lenguaje DAX

Los Cuatro Sabores de Cálculos DAX + Dos Postres y un Aroma



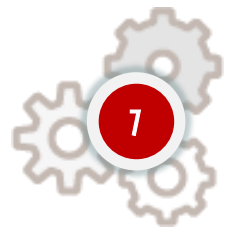
## Columnas Calculadas



Días de Entrega =

**INT** ( Pedidos[Fecha de Llegada] - Pedidos[Fecha de Pedido] )

País	Fecha de Pedido	Fecha de Envío	Fecha de Llegada	Días de Entrega
Colombia	06/04/2012	07/04/2012	21/04/2012	15
Argentina	07/04/2012	08/04/2012	20/04/2012	13
Argentina	41006	07/04/2012	41021	15
Colombia	08/04/2012	09/04/2012	04/05/2012	



# Introducción al Lenguaje DAX

Los Cuatro Sabores de Cálculos DAX + Dos Postres y un Aroma



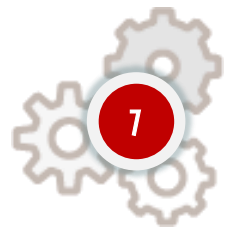
## Columnas Calculadas



Días de Entrega =

**INT** ( Pedidos[Fecha de Llegada] - Pedidos[Fecha de Pedido] )

País	Fecha de Pedido	Fecha de Envío	Fecha de Llegada	Días de Entrega
Colombia	06/04/2012	07/04/2012	21/04/2012	15
Argentina	07/04/2012	08/04/2012	20/04/2012	13
Argentina	07/04/2012	07/04/2012	22/04/2012	15
Colombia	41006	09/04/2012	41021	26



# Introducción al Lenguaje DAX

Los Cuatro Sabores de Cálculos DAX + Dos Postres y un Aroma



## Columnas Calculadas

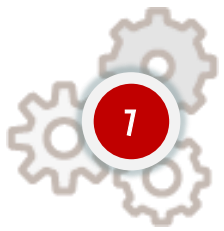


Días de Entrega =

**INT** ( Pedidos[Fecha de Llegada] - Pedidos[Fecha de Pedido] )

País	Fecha de Pedido	Fecha de Envío	Fecha de Llegada	Días de Entrega
Colombia	06/04/2012	07/04/2012	21/04/2012	15
Argentina	07/04/2012	08/04/2012	20/04/2012	13
Argentina	07/04/2012	07/04/2012	22/04/2012	15
Colombia	41006	09/04/2012	41021	26





# Introducción al Lenguaje DAX

Los Cuatro Sabores de Cálculos DAX + Dos Postres y un Aroma



## Columnas Calculadas



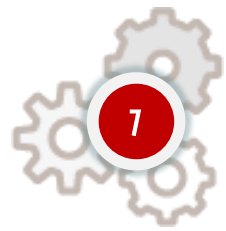
Días de Entrega =

**INT** ( Pedidos[Fecha de Llegada] - Pedidos[Fecha de Pedido] )

### CONTEXTO DE FILA AUTOMÁTICO O IMPLÍCITO

País	Fecha de Pedido	Fecha de Envío	Fecha de Llegada	Días de Entrega
Colombia	06/04/2012	07/04/2012	21/04/2012	15
Argentina	07/04/2012	08/04/2012	20/04/2012	13
Argentina	07/04/2012	07/04/2012	22/04/2012	15
Colombia	08/04/2012	09/04/2012	04/05/2012	26

El **Contexto de Fila Automático** es la restricción de las filas visibles a la cuales tiene acceso la expresión DAX, en particular es la restricción a la fila actual en análisis dejando. **“Oculta”** todas las demás filas en la tabla. El contexto de fila automático es generado por la creación de una columna calculada (*desde que se coloca el igual (=)*), y este itera desde la primera fila hasta la última ejecutando la expresión escalar definida.



# Introducción al Lenguaje DAX

*Función que Ignora el Contexto de Fila – Funciones de Agregación*

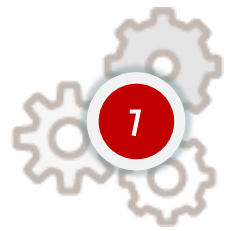


## Columnas Calculadas



Ingreso Total =  
SUM ( Pedidos[Ingresos] )

País	Costo Producto	Costo de Empaque	Costo de Envío	Ingresos	Ingresos Totales
Argentina	0,1	0,1	0,1	2	16
Colombia	0,2	0,1	0,1	3	
Argentina	0,2	0,2	0,1	2	
Brasil	0,1	0,1	0,1	2	
Colombia	0,1	0,1	0,1	4	
Colombia	0,2	0,1	0,2	3	



# Introducción al Lenguaje DAX

*Función que Ignora el Contexto de Fila – Funciones de Agregación*

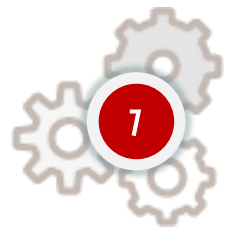


## Columnas Calculadas



Ingreso Total =  
SUM ( Pedidos[Ingresos] )

País	Costo Producto	Costo de Empaque	Costo de Envío	Ingresos	Ingresos Totales
Argentina	0,1	0,1	0,1	2	16
Colombia	0,2	0,1	0,1	3	16
Argentina	0,2	0,2	0,1	2	
Brasil	0,1	0,1	0,1	2	
Colombia	0,1	0,1	0,1	4	
Colombia	0,2	0,1	0,2	3	



# Introducción al Lenguaje DAX

*Función que Ignora el Contexto de Fila – Funciones de Agregación*

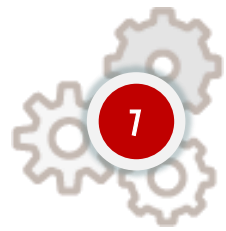


## Columnas Calculadas



Ingreso Total =  
SUM ( Pedidos[Ingresos] )

País	Costo Producto	Costo de Empaque	Costo de Envío	Ingresos	Ingresos Totales
Argentina	0,1	0,1	0,1	2	16
Colombia	0,2	0,1	0,1	3	16
Argentina	0,2	0,2	0,1	2	16
Brasil	0,1	0,1	0,1	2	
Colombia	0,1	0,1	0,1	4	
Colombia	0,2	0,1	0,2	3	



# Introducción al Lenguaje DAX

*Función que Ignora el Contexto de Fila – Funciones de Agregación*

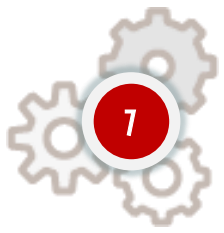


## Columnas Calculadas



Ingreso Total =  
SUM ( Pedidos[Ingresos] )

País	Costo Producto	Costo de Empaque	Costo de Envío	Ingresos	Ingresos Totales
Argentina	0,1	0,1	0,1	2	16
Colombia	0,2	0,1	0,1	3	16
Argentina	0,2	0,2	0,1	2	16
Brasil	0,1	0,1	0,1	2	16
Colombia	0,1	0,1	0,1	4	
Colombia	0,2	0,1	0,2	3	



# Introducción al Lenguaje DAX

*Función que Ignora el Contexto de Fila – Funciones de Agregación*

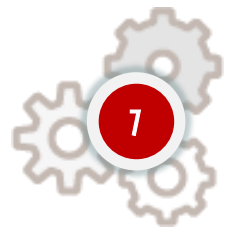


## Columnas Calculadas



Ingreso Total =  
SUM ( Pedidos[Ingresos] )

País	Costo Producto	Costo de Empaque	Costo de Envío	Ingresos	Ingresos Totales
Argentina	0,1	0,1	0,1	2	16
Colombia	0,2	0,1	0,1	3	16
Argentina	0,2	0,2	0,1	2	16
Brasil	0,1	0,1	0,1	2	16
Colombia	0,1	0,1	0,1	4	16
Colombia	0,2	0,1	0,2	3	



# Introducción al Lenguaje DAX

*Función que Ignora el Contexto de Fila – Funciones de Agregación*

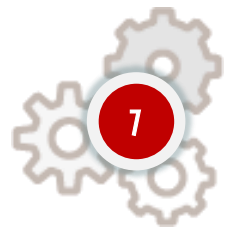


## Columnas Calculadas



Ingreso Total =  
SUM ( Pedidos[Ingresos] )

País	Costo Producto	Costo de Empaque	Costo de Envío	Ingresos	Ingresos Totales
Argentina	0,1	0,1	0,1	2	16
Colombia	0,2	0,1	0,1	3	16
Argentina	0,2	0,2	0,1	2	16
Brasil	0,1	0,1	0,1	2	16
Colombia	0,1	0,1	0,1	4	16
Colombia	0,2	0,1	0,2	3	16



# Introducción al Lenguaje DAX

*Función que Ignora el Contexto de Fila – Funciones de Agregación*



## Columnas Calculadas



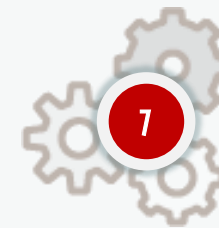
Ingreso Total =  
SUM ( Pedidos[Ingresos] )

País	Costo Producto	Costo de Empaque	Costo de Envío	Ingresos	Ingresos Totales
Argentina	0,1	0,1	0,1	2	16
Colombia	0,2	0,1	0,1	3	16
Argentina	0,2	0,2	0,1	2	16
Brasil	0,1	0,1	0,1	2	16
Colombia	0,1	0,1	0,1	4	16
Colombia	0,2	0,1	0,2	3	16

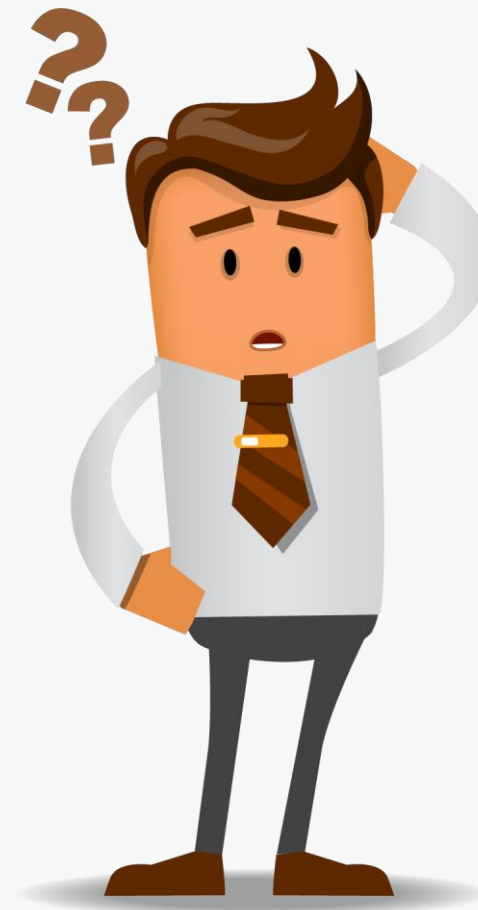


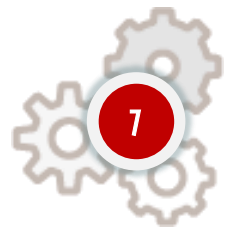
# Introducción al Lenguaje DAX

*Creando Columna Calculada con Función Lógica y de Función de Texto*



**Crear la Columna Calculada**  
**TipoDeEmbalaje:** Si el ID de Producto (SKU) inicia con la letra L (Libro), entonces asignar la palabra “Primario” de lo contrario, “Secundario”





# Introducción al Lenguaje DAX

Creando Columna Calculada con Función Lógica y de Función de Texto



## Columnas Calculadas

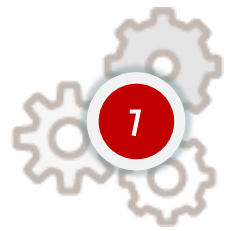


TipoDeEmbalaje =

IF ( LEFT ( Pedidos[ID] ) = "L", "Primario", "Secundario" )

B = L ×

País	ID	Tipo Compra	Ingresos	Tipo Empaque
Colombia	B01	Normal	30	Secundario
Argentina	B02	Normal	320	
Colombia	CBO1	Devolución	110	
Perú	L01	Normal	250	
Colombia	CC01	Normal	110	
Perú	L03	Devolución	250	



# Introducción al Lenguaje DAX

Creando Columna Calculada con Función Lógica y de Función de Texto



## Columnas Calculadas



TipoDeEmbalaje =

IF ( LEFT ( Pedidos[ID] ) = "L", "Primario", "Secundario" )

B = L ×

País	ID	Tipo Compra	Ingresos	Tipo Empaque
Colombia	B01	Normal	30	Secundario
Argentina	B02	Normal	320	Secundario
Colombia	CBO1	Devolución	110	
Perú	L01	Normal	250	
Colombia	CC01	Normal	110	
Perú	L03	Devolución	250	

# Introducción al Lenguaje DAX

Creando Columna Calculada con Función Lógica y de Función de Texto



## Columnas Calculadas



TipoDeEmbalaje =

IF ( LEFT ( Pedidos[ID] ) = "L", "Primario", "Secundario" )

C = L ×

País	ID	Tipo Compra	Ingresos	Tipo Empaque
Colombia	B01	Normal	30	Secundario
Argentina	B02	Normal	320	Secundario
Colombia	CBO1	Devolución	110	Secundario
Perú	L01	Normal	250	
Colombia	CC01	Normal	110	
Perú	L03	Devolución	250	

# Introducción al Lenguaje DAX

Creando Columna Calculada con Función Lógica y de Función de Texto



## Columnas Calculadas

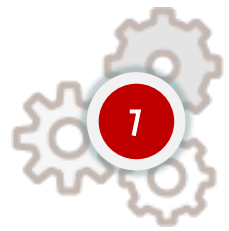


TipoDeEmbalaje =

IF ( LEFT ( Pedidos[ID] ) = "L", "Primario", "Secundario" )

L = L ✓

País	ID	Tipo Compra	Ingresos	Tipo Empaque
Colombia	B01	Normal	30	Secundario
Argentina	B02	Normal	320	Secundario
Colombia	CBO1	Devolución	110	Secundario
Perú	L01	Normal	250	Primario
Colombia	CC01	Normal	110	
Perú	L03	Devolución	250	



# Introducción al Lenguaje DAX

Creando Columna Calculada con Función Lógica y de Función de Texto



## Columnas Calculadas

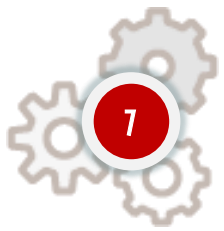


TipoDeEmbalaje =

IF ( LEFT ( Pedidos[ID] ) = "L", "Primario", "Secundario" )

C = L ×

País	ID	Tipo Compra	Ingresos	Tipo Empaque
Colombia	B01	Normal	30	Secundario
Argentina	B02	Normal	320	Secundario
Colombia	CBO1	Devolución	110	Secundario
Perú	L01	Normal	250	Primario
Colombia	CC01	Normal	110	Secundario
Perú	L03	Devolución	250	



# Introducción al Lenguaje DAX

Creando Columna Calculada con Función Lógica y de Función de Texto



## Columnas Calculadas



TipoDeEmbalaje =

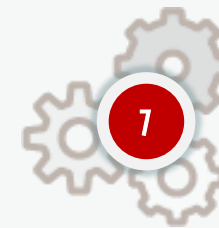
IF ( LEFT ( Pedidos[ID] ) = "L", "Primario", "Secundario" )

C = L ✓

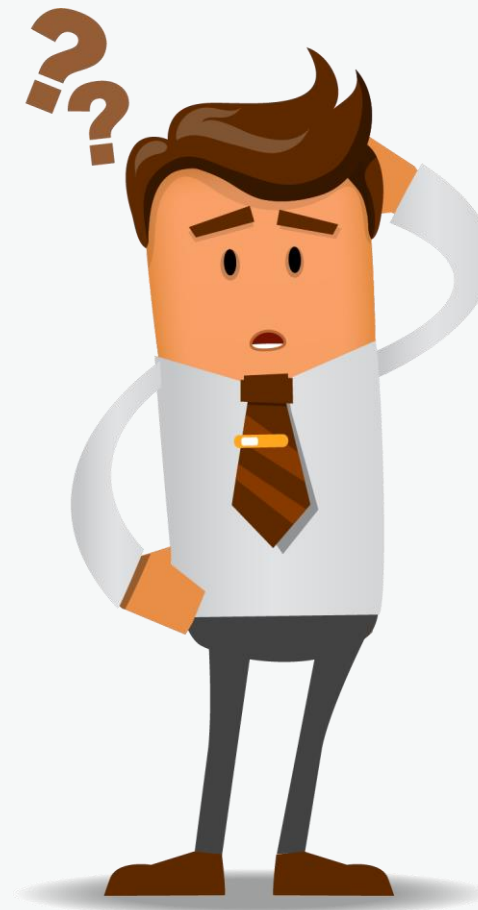
País	ID	Tipo Compra	Ingresos	Tipo Empaque
Colombia	B01	Normal	30	Secundario
Argentina	B02	Normal	320	Secundario
Colombia	CBO1	Devolución	110	Secundario
Perú	L01	Normal	250	Primario
Colombia	CC01	Normal	110	Secundario
Perú	L03	Devolución	250	Primario

# Introducción al Lenguaje DAX

*Creando Columna Calculada con Función Lógica y de Función de Texto*



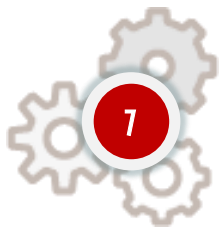
**Crear la Columna Calculada**  
**TipoDeEmbalaje:** Si el ID de Producto (SKU) inicia con la letra L (Libro) o B (BlueRay), entonces asignar la palabra “Primario” de lo contrario, “Secundario”





# Introducción al Lenguaje DAX

*Los Cuatro Sabres de Cálculos DAX + Dos Postres y un Aroma*



## COLUMNAS CALCULADAS

Una columna calculada como su nombre lo indica es un nuevo campo que se agrega en una tabla existente en el modelo de datos, creada a partir (aunque no es obligatorio) de columnas nativas.

→ Se almacenan en memoria (in-memory), dicho de otro modo, consumen preciosa memoria RAM.

→ Se recalculan en el momento de actualizar el reporte, teniendo que re ejecutar sus iteraciones.

→ Trabaja en un **Contexto de Fila**, dicho contexto de fila se genera automáticamente por la C.C.

→ Trabaja en un **Contexto de Filtro**, aunque este no es evidente al principio porque esta vacío.

→ Fáciles de utilizar e inofensivas para cardinalidad de columnas baja y pocas columnas calculadas.

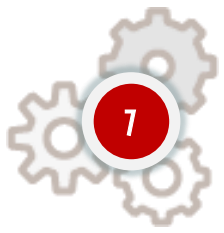
→ Se visualizan en la sección de informes y en la sección de datos en la interfaz.

# Sobre VARIABLES

The background features a dark blue color scheme with various financial data visualizations. On the left, there are vertical columns of numbers including .3, 6.03, .25, 1.0, 1.3, 1, and 03. In the center and right, there are faint, semi-transparent charts: a candlestick chart with an upward-pointing arrow, a bar chart, and a line graph with a circular data point. The overall aesthetic is technical and data-driven.

# Introducción al Lenguaje DAX

Los Tres + Dos Sabores de Cálculos DAX más un Aroma



## VARIABLES

Son “*Líneas de Código*” que nos permiten utilizar el resultado de una expresión bien sea escalar o tabular, como entrada en otras partes de una expresión DAX. Se debe tener presente que una vez calculada el valor de una variable se mantiene fijo y no cambia.

→ Cuando creamos una variable **SÓLO existe en la expresión** y no puede ser utilizada en otra.

→ La palabra clave **VAR** introduce la definición de la variable; y **RETURN** define la expresión que será retornada.

→ **Facilitan la lectura** de las expresiones, por eso es recomendable como buena practica.

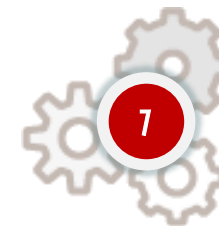
→ Si distintas secciones de una expresión se repiten, el uso de variables **optimiza el calculo** en el Motor DAX.

→ Las variables hacen ver el código mas grande, pero es más legible y su aplicación tiene múltiples beneficios.

→ Permite dividir problemas complejos en pequeñas tareas a resolver para conseguir un objetivo mayor.

# Introducción al Lenguaje DAX

Formateo de Expresiones DAX y DAXFormatter.Com



- 1) Si se decide dividir los parámetros de una función con saltos de línea, estos deben ir indentados (sangrías) un nivel más respecto al nombre de la función
- 2) El paréntesis de cierre de la función debe ir en su propia línea y alineado con el nombre de la función.

```
IF (  
    LEFT ( Pedidos[SKU] ) = "L",  
    "Primario",  
    "Secundario"  
)
```

- 3) Nunca usar la sintaxis abreviada de CALCULATE: [Medida] ( Filtro ). En su lugar, se debe usar CALCULATE ( [Medida], Filtro ).
- 4) Siempre poner un espacio antes de los paréntesis '(' y ')'.  
5) Siempre poner un espacio antes de cualquier operando y operador en una expresión.
- 6) Nunca poner un espacio entre el nombre de la tabla y el nombre de la columna.

[Ingresos Tot][SKU]Productos [SKU]="CB01")+10

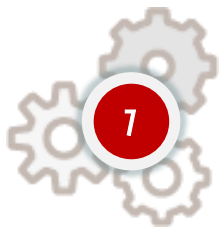


CALCULATE ( [Ingresos Tot], SKUProductos[SKU] = "CB01" ) + 10



# Introducción al Lenguaje DAX

Formateo de Expresiones DAX y DAXFormatter.Com



- 7) Siempre usar nombres de tabla para referencias de columna, incluso al definir una columna calculada dentro de una tabla, Ej: [País] mejor Pedidos[País].
- 8) Nunca usar el nombre de una tabla cuando se hace referencia a medidas, Ej: Medidas[Ingresos Tot] mejor [Ingresos Tot].
- 9) Para columna calculadas temporales anteponga un @ a excepción que sea una columna temporal que se proyecta como resultado en una consulta o tabla culada.



**DAX Formatter**

<https://www.daxformatter.com/>



**BRAVO**

<https://bravo.bi/>

# Tablas Calculadas

# Función de Tabla: VALUES

Categoría: Manipulación de Tablas – Atributo: --

6



TABLA

## VALUES

Una Tabla Completa Con Una o Más Columnas



### COMPATIBILIDAD

Microsoft Excel  
Excel ≥ 2013

★★★★


Power BI Desktop  
PBI ≥ Nov 2016

SSAS Tabular  
SSAS ≥ 2012

★★★★

Azure Analysis  
Actual

## DESCRIPCIÓN

La función **VALUES** recibe una columna y devuelve los valores únicos con fila **BLANK** por violación a referencia integral si aplica; o recibe una tabla en cuyo caso la retorna igual pero con fila **BLANK** si hay violación de la referencia integral.  [Más Información Aquí](#)

## SINTAXIS

VALUES ( < Tabla > )

### I TABLA

Referencia a una columna en una tabla en el modelo de datos, ejemplo: *Pedidos[País]*o

Referencia a una tabla en el modelo de datos , ejemplos: *Pedidos*.

### Tipo

Obligatorio

### Atributo

No Repetible 

# Función de Tabla: SELECTCOLUMNS

Categoría: Manipulación de Tablas – Atributo: Iteración

6



TABLA

## SELECTCOLUMNS

Una Tabla Completa Con Una o Más Columnas



### COMPATIBILIDAD

Microsoft Excel  
Excel ≥ 2016

★★★★★


Power BI Desktop  
PBI ≥ Nov 2016

SSAS Tabular  
SSAS ≥ 2016

★★★★★

Azure Analysis  
Actual

## DESCRIPCIÓN

La función [SELECTCOLUMNS](#) devuelve las columnas que se indiquen dada una tabla o expresión de tabla (*es decir anidación o composición de tablas*), adicionalmente, se pueden generar nuevas columnas extendidas (*columnas calculadas*) indicando una expresión de tipo escalar que se ejecuta en base fila a fila.  [Más información aquí](#)

## SINTAXIS

1 Primer Parámetro      Segundo Parámetro (Opcional) 2

```
SELECTCOLUMNS ( < Tabla > , < Nombre > , < Expresión > , [ < Nombre > , < Expresión > ] , [ ... ] )
```

3 Tercer Parámetro

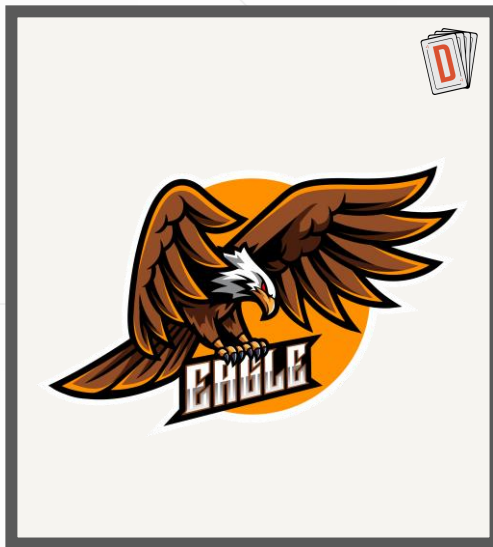
1	<b>TABLA</b>	Tabla para seleccionar columnas.	<i>Obligatorio / No Repetible</i>
2	<b>NOMBRE</b>	Nombre de la columna ser añadida.	<i>Obligatorio / Repetible</i>
3	<b>EXPRESIÓN</b>	Expresión para la nueva columna para añadir.	<i>Obligatorio / Repetible</i>



# Función de Tabla: CALENDAR

Categoría: Manipulación de Tablas – Atributo: --

6



TABLA

## CALENDAR

Una Tabla Completa Con Una o Más Columnas



### COMPATIBILIDAD

Microsoft Excel  
Excel ≥ 2016

★★★★


Power BI Desktop  
PBI ≥ Nov 2016

SSAS Tabular  
SSAS ≥ 2016

★★★★

Azure Analysis  
Actual

## DESCRIPCIÓN

La función **CALENDAR** retorna una tabla de una sola columna de con las fechas consecutivas dada una fecha inicial y una fecha final, estas fechas se indican comúnmente con el siguiente formato: "YYYY-MM-DD" o con el payo de DATE así: **DATE** ( *Año , Mes , Día* )  [Más información aquí](#)

## SINTAXIS

1 Primer Parámetro

CALENDAR ( < Fecha Inicial > , < Fecha Final > )

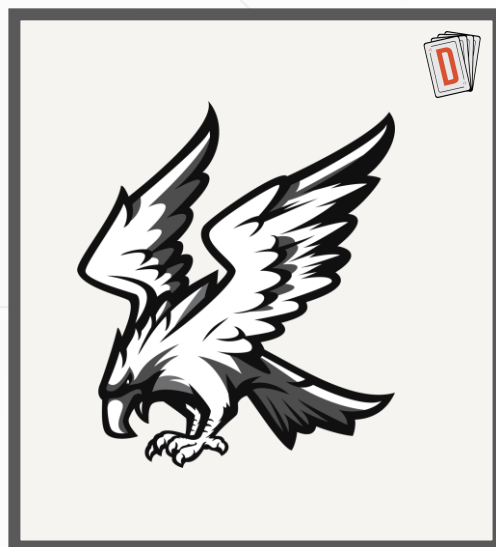
2 Tercer Parámetro

1	<b>FECHA INICIAL</b>	Día de partida en tipo de dato fecha	<i>Obligatorio / No Repetible</i>
2	<b>FECHA FINAL</b>	Día de final en tipo de dato fecha	<i>Obligatorio / No Repetible</i>

# Función de Tabla: CALENDAR

Categoría: Manipulación de Tablas – Atributo: --

6



TABLA

## CALENDAR

Una Tabla Completa Con Una o Más Columnas



### COMPATIBILIDAD

Microsoft Excel  
Excel ≥ 2016

★★★★★


Power BI Desktop  
PBI ≥ Nov 2016

SSAS Tabular  
SSAS ≥ 2016

★★★★★

Azure Analysis  
Actual

## DESCRIPCIÓN

La función **CALENDARAUTO** retorna una tabla de una sola columna de tipo fecha arranco desde la fecha más pequeña en el modelo hasta la más grande de manera consecutiva. En su único parámetro opcional se puede indicar la última fecha del año (especialmente útil para el año fiscal).  [Más información aquí](#)

## SINTAXIS

1 Primer Parámetro

CALENDARAUTO ( < Fin del Año Fiscal > )

2 Tercer Parámetro

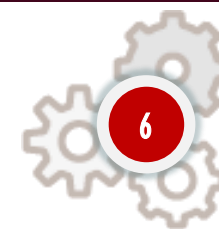
1 FIN AÑO

Mes número final del año fiscal

Opcional / No Repetible

# Construcción de Tablas de Calendario

Columnas Calculadas Típicas para la Tabla de Calendario



0. Fecha = CALENDARAUTO ( )

1. Año = YEAR ( Calendario[Fecha] )

2. Trimestre = QUARTER ( Calendario[Fecha] ) 

3. Trimestre Prefijo = "Trimestre " & QUARTER ( Calendario[Fecha] )

4. Mes Número = MONTH ( Calendario[Fecha] )

5. Mes Nombre = FORMAT ( Calendario[Fecha] , "mmmm" )

6. DS Número = WEEKDAY ( Calendario[Fecha] , 2 )

7. DS Nombre = FORMAT ( Calendario[Fecha] , "dddd" )

8. Semana = WEENUM ( Calendario[Fecha] )

Power BI anterior a Diciembre del 2019

A. Trimestre = FORMAT ( Calendario[Fecha] , "q" )

Power Pivot para Excel

B. Trimestre = ROUNDUP ( MONTH ( Calendario[Fecha] ) / 3 , 0 )



Emplear la función **CALENDAR** si existen columnas de fecha que agreguen más filas que no son de interés para el análisis, ejemplo: Fecha de cumpleaños.



**Recetas DAX en Cartas DAX**

[→ Visitar Aquí ←](#)

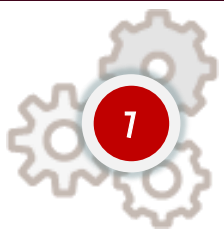


**BRAVO para Power BI**

[→ Descargar Aquí ←](#)

# Introducción al Lenguaje DAX

*Los Tres + Dos Sabores de Cálculos DAX más un Aroma*



## TABLAS CALCULADAS

Una tabla calculada como su nombre lo indica es una nueva tabla que se agrega al modelo de datos creada a partir (aunque no es obligatorio) de tablas nativas en el propio modelo.

→ Se almacenan en memoria (in-memory), dicho de otro modo, consumen preciosa memoria RAM.

→ Se recalculan en el momento de actualizar el reporte, teniendo que re ejecutar sus iteraciones.

→ Puede tener un **Contexto de Fila**, aunque no necesariamente (No por C.C sino por funciones de iteración).

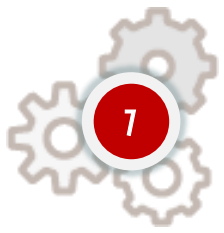
→ Trabaja en un **Contexto de Filtro**, aunque este no es evidente al principio porque esta vacío.

→ Fáciles de utilizar e inofensivas para pocas tablas y derivadas de pequeñas tablas de búsqueda.

→ Se visualizan en la sección de informes y en la sección de datos en la interfaz.

# Bibliografía y Recursos para Complementar

Videos, Artículos y Más



## CONSTRUCCIÓN DE TABLAS DE CALENDARIO Y CÁLCULOS DE INTELIGENCIA DE TIEMPO



*“En este seminario vemos tres formas de crear la tabla de Calendario con detalles, además, de estudiar las funciones de inteligencia de tiempo, es decir, aquellas funciones que nos permiten trabajar con periodos de tiempo como: año, trimestre, mes y día para determinar métricas acumulados, comparación respecto al año, seguimientos, etc.”*



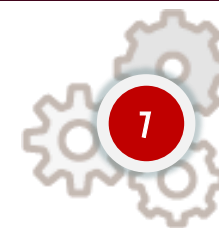
**ESTUDIAR VÍDEO AQUÍ | 2:44:14**

# Medidas

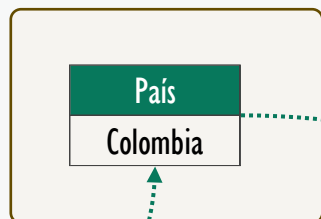


# Medidas en el Lenguaje y Contexto de Filtro

Los Tres Pasos Primordiales en DAX, Lo Más Importante



## CONTEXTO DE FILTRO



País	%Gnc_M
Argentina	
Brasil	
Chile	
Colombia	174%
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

País	SKU	Ingresos	Utilidad	%Gan_M
Colombia	L01	30	14	47%
Argentina	CB01	320	290	91%
Colombia	CB01	110	70	64%
Perú	L01	250	200	80%
Colombia	CC01	110	70	64%
Perú	CB01	250	200	80%

## Motor DAX



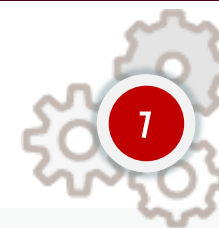
| Crea la Medida |

`__%Gan_M :=  
SUM ( Pedidos[%Gnc_M] )`

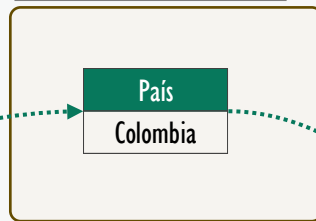


# Medidas en el Lenguaje y Contexto de Filtro

Los Tres Pasos Primordiales en DAX, Lo Más Importante



CONTEXTO DE FILTRO




País	%Gnc_B
Argentina	
Brasil	
Chile	
Colombia	0,22
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

$$\%Gnc\_B := \text{DIVIDE} \left( \text{SUM} (\text{Pedidos}[\text{Utilidad}]), \text{SUM} (\text{Pedidos}[\text{Ingresos}]) \right), \frac{2}{9} = 0,22$$

## IMPORTANT NOTICE

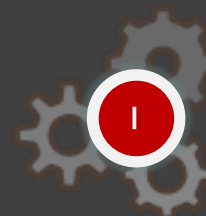
Piensa; siempre, Siempre, SIEMPRE, en términos de los tres pasos primordiales en DAX: Identificar, Aplicar Ejecutar

1  Identificar Filtros

2  Aplicar Filtros

3  Ejecutar Expresión DAX





## Medidas

Cálculos que tienen como fin ejecutarse en subconjunto de filas dado un conjunto de restricciones (contexto de filtro), por ello, es un tipo de cálculo que van directamente al área de valores, por lo tanto, creados a partir de hechos.



## Medidas Implícitas

Al arrastrar una columna al área de valores, automáticamente Power BI le asigna internamente una expresión DAX, si es numérico la suma, si es texto el primer valor o el recuento, empero, no tenemos manera de acceder a dicha fórmula.



## Medidas Explícitas

Son las expresiones para el área de valores creadas por nosotros de forma "manual", por lo tanto, tenemos acceso a su fórmula, la podemos modificar y adicionalmente podemos llamar dicha medida en otras medidas.



## Medidas Manuales

Estas expresiones son elaboradas 100% con nuestro conocimiento del lenguaje DAX, desde cero sin ninguna receta.



## Medidas Automáticas

Estas expresiones son elaboradas automáticamente, y dejan una expresión que podemos modificar.



## Medidas Rápida

Estas expresiones son elaboradas automáticamente con una receta, y dejan una expresión DAX.

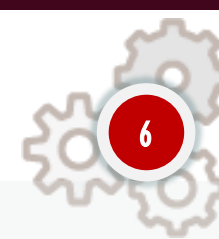


## Medidas Copilot

Estas expresiones son elaboradas mediante inteligencia artificial, con interacción al estilo Chat-GPT.

# Medidas en el Lenguaje y Contexto de Filtro

LA CLAVE: Los Tres Pasos Primordiales en DAX.



## CONTEXTO DE FILTRO (Original)



País	Ingresos Tot
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	7
Uruguay	
Venezuela	
<b>Total</b>	

1 Identificar Filtros

País	Tipo Compra	Ingresos
Argentina	Normal	2
Argentina	Normal	3
Argentina	Normal	2
Colombia	Normal	2
Colombia	Normal	4
Colombia	Normal	
Perú	Normal	1
Perú	Normal	2
Perú	Normal	4

$\Sigma = 7$

2 Aplicar Filtros

Ingresos Tot =  
SUM ( Pedidos[Ingresos] )

## CONCLUSIONES DE LA EXPRESIÓN

- La expresión se ejecutó en el pedacito de tabla visible, gracias a SUM.
- No todas las funciones respetan el contexto de filtro, ALL\* y LOOKUPVALUE

## CONCLUSIONES DEL VALOR DE RETORNO

- El valor devuelto en la casilla de análisis "verde" debe ser un valor escalar (único)
- Cada celda en la matriz se ejecuta uno a uno en un proceso iterativo.

3 Ejecutar Expresión DAX

# Medidas en el Lenguaje y Contexto de Filtro

LA CLAVE: Los Tres Pasos Primordiales en DAX.



## Las Personas Lo Olvidan



CONTEXTO DE FILTRO (Normal)



País	Ingresos Tot
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	7
Uruguay	
Venezuela	
<b>Total</b>	

1

Identificar Filtros

## Las Personas Lo Olvidan



País	Categoría	Ingresos
Argentina		2
Argentina		3
Argentina		2
Colombia	Normal	2
Colombia	Normal	2
Colombia	Normal	2
Perú	Normal	1
Perú	Normal	2
Perú	Normal	4

 $\Sigma = 7$ 

2

Aplicar Filtros

$$\text{Ingresos Tot} = \text{SUM}(\text{Pedidos}[\text{Ingresos}])$$

### CONCLUSIONES DE LA EXPRESIÓN

→ La expresión se ejecutó en el pedacito de tabla visible, gracias a SUM.

→ No todas las funciones respetan el contexto de filtro, ALL\* y LOOKUPVALUE

### CONCLUSIONES DEL VALOR DE RETORNO

→ El valor devuelto en la casilla de análisis "verde" debe ser un valor escalar (único)

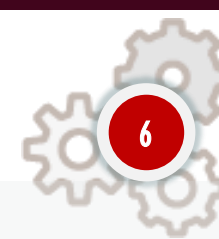
→ Cada celda en la matriz se ejecuta uno a uno en un proceso iterativo.

3

Ejecutar Expresión DAX

# Medidas en el Lenguaje y Contexto de Filtro

LA CLAVE: Los Tres Pasos Primordiales en DAX.



## Las Personas Lo Olvidan



CONTEXTO DE FILTRO (Normal)



País	Ingresos Tot
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	7
Uruguay	
Venezuela	
<b>Total</b>	

1

Identificar Filtros

## Las Personas Lo Olvidan



País	Categoría	Ingresos
Perú	Normal	1
Perú	Normal	2
Perú	Normal	4
		$\Sigma = 7$

2

Aplicar Filtros



Ingresos Tot =  
SUM ( Pedidos[Ingresos] )

## CONCLUSIONES DE LA EXPRESIÓN

- La expresión se ejecutó en el pedacito de tabla visible, gracias a SUM.
- No todas las funciones respetan el contexto de filtro, ALL\* y LOOKUPVALUE

## CONCLUSIONES DEL VALOR DE RETORNO

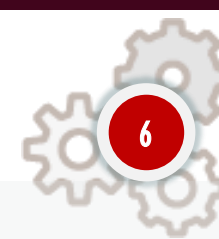
- El valor devuelto en la casilla de análisis "verde" debe ser un valor escalar (único)
- Cada celda en la matriz se ejecuta uno a uno en un proceso iterativo.

3

Ejecutar Expresión DAX

# Medidas en el Lenguaje y Contexto de Filtro

LA CLAVE: Los Tres Pasos Primordiales en DAX.



## Las Personas Lo Olvidan



CONTEXTO DE FILTRO (Normal)



País	Ingresos Tot
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	7
Uruguay	
Venezuela	
<b>Total</b>	

1

Identificar Filtros

## Las Personas Lo Olvidan



País	Categoría	Ingresos
Perú	Normal	1
Perú	Normal	2
Perú	Normal	4
		$\Sigma = 7$

2

Aplicar Filtros



Ingresos Tot =  
SUM ( Pedidos[Ingresos] )

## CONCLUSIONES DE LA EXPRESIÓN

- La expresión se ejecutó en el pedacito de tabla visible, gracias a SUM.
- No todas las funciones respetan el contexto de filtro, ALL\* y LOOKUPVALUE

## CONCLUSIONES DEL VALOR DE RETORNO

- El valor devuelto en la casilla de análisis "verde" debe ser un valor escalar (único)
- Cada celda en la matriz se ejecuta uno a uno en un proceso iterativo.

3

Ejecutar Expresión DAX

# Evaluando Tu Entendimiento

Los Tres Pasos Primordiales en DAX

7

Tomemos la siguiente matriz: **E** imaginemos que se desea retorna 1 en la celda de Perú y 2 en todas las demás.

País	UnoDos
Argentina	2
Brasil	2
Chile	2
Colombia	2
Ecuador	2
Paraguay	2
Perú	1
Uruguay	2
Venezuela	2
<b>Total</b>	<b>2</b>

La siguiente solución se pone sobre la mesa con frecuencia:

```
=IF ( Pedidos[País] = "Perú" , 1 , 2 )
```

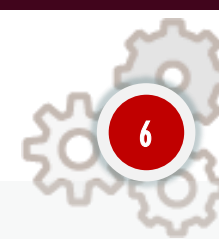
¿LA EXPRESIÓN NO ES VÁLIDA ?



**CASI CAES EN LA TRAMPA**

# Medidas en el Lenguaje y Contexto de Filtro

LA CLAVE: Los Tres Pasos Primordiales en DAX.



CONTEXTO DE FILTRO (Original)

País

Perú

País	Ingresos Tot
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	
<b>Total</b>	

1

Identificar Filtros

Ese conjunto de valores, aunque sea el mismo, se debe convertir en un escalar.

País	Tipo Compra	Ingresos
Argentina	Normal	2
Argentina	Normal	3
Argentina	Normal	2
Colombia	Normal	2
Colombia	Normal	4
Colombia	Normal	3
Perú	Normal	1
Perú	Normal	2
Perú	Normal	4

2

Aplicar Filtros



UnoDos =

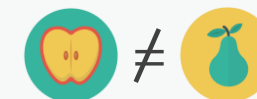
```
IF (
  Pedidos[País] = "Perú",
  1,
  2
)
```

Analizando la Prueba Lógica en el IF:

Perú
Perú
Perú

= "Perú"

¿Cómo vas a comparar una tabla con un valor Escalar?

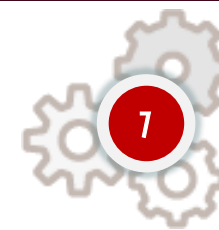


3

Ejecutar Expresión DAX

# Introducción al Lenguaje DAX

*Los Tres + Dos Sabores de Cálculos DAX más un Aroma*



## MEDIDAS

Las medidas son un tipo de cálculo destinadas a determinarse en un subconjunto de filas de una o más tablas en el modelo de datos de forma directa. Las medidas se ejecutan en ese subconjunto de filas gracias al contexto de filtro *(que es dictado inicialmente por elementos del objeto visual)*. Las medidas también pueden tener un carácter iterativo al estilo de las columnas calculadas, sin embargo, esto es dictado por funciones particulares del Lenguaje DAX.

→ No se almacena en memoria, por lo tanto, no consume ni una pizca de memoria RAM ni similares.

→ Se calcula *“on the fly”*, es decir, cuando se utilizan en un objeto visual en el momento de ejecución.

→ **Consume CPU** en el momento de utilización, por lo que se entreve, que si no se usan no gastan recursos.

→ Trabaja en un **Contexto de Filtro**, dicho elemento se genera automáticamente por elementos de la interfaz.

→ Puede tener un **Contexto de Fila**, aunque no necesariamente (No por C.C sino por funciones de iteración).

→ Se visualizan en la sección de informes y en la sección de datos en la interfaz.

→ Las medidas **van únicamente al área de valores**, no puede ir en filas, columnas, etc..



# ¡Muchas Gracias!

© Escuela de Inteligencia de Negocios / Excel Free Blog / Power Skill

Este Material Fue creado por:

Ing. Miguel A. Caballero Sierra

Ing. Fabian E. Torres Hernández

Por favor, mantenga este material para su estudio personal

