

Día 3 — En Vivo

A stylized phoenix logo with a book. The phoenix is depicted with orange and red flames, and the book is shown in a 3D perspective with a metallic, silver-colored frame. The background of the slide features a dark, atmospheric image of a castle or fortress on a rocky shore under a cloudy sky.

# SEMANA DE DAX

Día 3: Métricas de Agregación y de Razones



Escuela de Inteligencia de Negocios | [ExcelFreeBlog.com](https://www.ExcelFreeBlog.com)



**PREVIO A DAX:** La idea del lenguaje DAX es trabajar con múltiples tablas relacionadas, y para ello debemos distinguir dos conceptos un par de conceptos fundamentales para poder crear las relaciones: tablas transaccionales y tablas de búsqueda

**Tabla Transaccional:** Es una tabla que contiene registros en intervalos regulares como: día a día, semana a semana, etc. De un proceso de negocio y sus variables.

**Clave Externa:** Columna donde los valores repiten los valores y tiene equivalente.

**Tabla de Búsqueda:** Es una tabla que permite agrupar y restringir datos de una tabla transaccional, son tablas que tiene datos de los cuales dependen otros datos.

**Clave Primaria:** Columna donde los valores no se repiten y tiene equivalente.



**EVITAR:** relacionar tablas transaccionales con tablas transaccionales y por compatibilidad evitar relacionar tabla de búsqueda con tabla de búsqueda, es decir, **LO RECOMENDABLE** es relacionar tablas de búsqueda con tablas transaccionales. *Evitamos ambigüedad y bajo rendimiento.*

## 💀 Tipos de Relaciones:

- 💀 Uno a Muchos – Power BI, SSAS y Power Pivot.
- 💀 Uno a Uno – Power BI y SSAS.
- 💀 Muchos a Muchos – Power BI y SSAS. 🧐



**Nota:** Es posible tener más de una relación entre un par de tablas, sin embargo, una y solo una estará activa a la vez. (CALCULATE/USERELATIONSHIP) [\[Más aquí\]](#).



## Regla Dorada No.1:

Utilizar en el área de valores campos(columnas) de las tablas transaccionales, en todas las demás áreas campos de las tablas de búsqueda.

## Repaso y Síntesis

# Cálculos del Lenguaje DAX

Contexto de Filtro + Contexto de Fila



**ACERCA DEL LENGUAJE DAX:** En DAX no existe el concepto de celda individual, dado que sólo se trabaja con tablas completas y columnas completas, eso estriba en que su sintaxis es sencilla.

🔗 **TABLA:** Para llamar una tabla sólo basta con indicar su nombre.

- = Pedidos
- = 'Producción'
- = 'Ventas Globales'
- = 'Date'

🔗 **COLUMNA:** Para llamar una columna, nombre de tabla y columna en corchetes

- = Pedidos[SKU]
- = Pedidos[País]
- = Pedidos[Costo de Empaque]
- = Pedidos[Costo de Envío]



**ADVERTENCIA:** Aunque se puede llamar a una columna sin indicar en cual tabla reside, es decir, solo implementando los corchetes, ejemplo: [País], [SKU], [Costo de Empaque], etc. Esto es una muy mala práctica, para discernir de una operación llamada transición de contextos.

## → ¿En Dónde Utilizamos DAX? | Tres Sabores Cálculos DAX ←

### 1 Columnas Calculadas

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 (Consumen RAM).
- 🔗 Se recalcula al actualizar el reporte.
- 🔗 Trabajan en el **contexto de fila** principalmente
- 🔗 Se visualizan en la sección de informes.

**Variables:** Nombres para almacenar cálculos intermedios, los cuales ayudan con legibilidad y optimización. [\[Más aquí\]](#)

### 2 Tablas Calculadas

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 (Consumen RAM).
- 🔗 Se recalcula al actualizar el reporte.
- 🔗 Trabajan en el **contexto de fila** y **contexto de filtro**.
- 🔗 Se visualizan en la sección de informes.

### 3 Medidas

Es un tipo de cálculo que se determina en un subconjunto de filas en una o más tablas forma directa, a diferencia de una columna calculada que se ejecuta fila a fila.

- 🔗 No se almacena en memoria, no consumen RAM ni similares
- 🔗 Se calcula cuando se utilizan en un objeto visual.
- 🔗 Consumen CPU en el momento de utilización.
- 🔗 Trabaja en el **Contexto de Filtro** comúnmente aunque no exclusivamente

# Cálculos del Lenguaje DAX

Contexto de Filtro + Contexto de Fila



## Medidas

Cálculos que van directamente al área de valores, por lo tanto, creados de tablas transaccionales como buena practica.



## Medidas Implícitas

Al arrastrar una columna al área de valores, automáticamente Power BI le asigna internamente una expresión DAAX, si es numérico la suma, si es texto el recuento, sin embargo, no tenemos manera de acceder a dicha formula.



## Medidas Explicitas

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 Rápidas](#) [\[Más aquí\]](#)

# Clasificación de Funciones en el Lenguaje DAX

*Según Resultado, Comportamiento y Proceso*



## CLASIFICACIÓN DE FUNCIONES

Las funciones del lenguaje DAX se pueden clasificar de muchas maneras, es decir, se pueden crear categorías de acuerdo a diversos criterios u atributos, algunas clasificaciones alternas son:

- De acuerdo al valor que retornan      Según Resultado
- De acuerdo a su interacción con el contexto de evaluación      Según comportamiento respecto al contexto (RC).
- De acuerdo a su comportamiento interno      Según el proceso

# Clasificación de Funciones en el Lenguaje DAX

Según Resultado, Comportamiento y Proceso



## Según Resultado

Esta forma de clasificación de las funciones del lenguaje DAX consiste en poder identificar fácilmente si la función devuelve como resultado final un valor único o atómico, un conjunto de valores o tabla de datos, o absolutamente nada.

- **Escalar:** Función que devuelve un solo valor, no importa de que tipo (BINARY, BOOLEAN, DECIMAL, INTEGER, STRING, DATETIME), ejemplo: 10, TRUE, "Power BI", ...

### FUNCIONES DE AGREGACIÓN

#### Funciones de No-Conteo:

AVERAGE  
COUNT  
GEOMEAN  
MEDIAN  
MAX  
MIN  
PRODUCT  
PERCENTILE.EXC  
PERCENTILE.INC

STDEV.S  
STDEV.P  
VAR.P  
VAR.S  
SUM

AVERAGEA  
COUNTA  
MAXA  
MINA

No Útiles

#### Funciones de Conteo:

COUNTBLANK  
COUNTROWS  
DISTINCTCOUNT  
DISTINCTCOUNTBLANK  
APPROXIMATEDISTINCTCOUNT

### FUNCIONES DE NO-AGREGACIÓN

#### Funciones Lógicas:

AND  
FALSE  
IF  
IFERROR  
NOT  
OR  
SWITCH  
TRUE  
COALESCE

#### Funciones Matemáticas:

DIVIDE  
ROUNDUP  
ROUNDDOWN  
SQRT  
RAND  
RANDBETWEEN  
SIN  
COS

#### Funciones de Fecha:

DATE  
MONTH  
QUARTER  
NOW  
RAND  
WEEKDAY  
WEEKNUM  
YEAR

Diciembre 2019

Marzo 2020

# Clasificación de Funciones en el Lenguaje DAX

Según Resultado, Comportamiento y Proceso



## Según Resultado

Esta forma de clasificación de las funciones del lenguaje DAX consiste en poder identificar fácilmente si la función devuelve como resultado final un valor único o atómico, un conjunto de valores o tabla de datos, o absolutamente nada.

- **Escalar:** Función que devuelve un solo valor, no importa de que tipo (BINARY, BOOLEAN, DECIMAL, INTEGER, STRING, DATETIME), ejemplo: 10, TRUE, "Power BI", ...

### FUNCIONES DE AGREGACIÓN

#### Funciones de No-Conteo:

AVERAGE  
COUNT  
GEOMEAN  
MEDIAN  
MAX  
MIN  
PRODUCT  
PERCENTILE.EXC  
PERCENTILE.INC

STDEV.S  
STDEV.P  
VAR.P  
VAR.S  
SUM

AVERAGEA  
COUNTA  
MAXA  
MINA

No Útiles

#### Funciones de Conteo:

COUNTBLANK  
COUNTROWS  
DISTINCTCOUNT  
DISTINCTCOUNTBLANK  
APPROXIMATEDISTINCTCOUNT

#### Funciones Lógicas:

AND  
FALSE  
IF  
IFERROR  
NOT  
OR  
SWITCH  
TRUE  
COALESCE

Marzo 2020

### FUNCIONES DE NO-AGREGACIÓN

#### Funciones Matemáticas:

DIVIDE  
ROUNDUP  
ROUNDDOWN  
SQRT  
RAND  
RANDBETWEEN  
SIN  
COS

#### Funciones de Fecha:

DATE  
MONTH  
QUARTER  
NOW  
RAND  
WEEKDAY  
WEEKNUM  
YEAR

Diciembre 2019

# Clasificación de Funciones en el Lenguaje DAX

Según Resultado, Comportamiento y Proceso



## Según Resultado

Esta forma de clasificación de las funciones del lenguaje DAX consiste en poder identificar fácilmente si la función devuelve como resultado final un valor único o atómico, un conjunto de valores o tabla de datos, o absolutamente nada.

- **Tabular:** Función que devuelve un conjunto de valores en formato de tabla, es decir, es un arreglo de valores escalares.

<u>Funciones de Listas:</u>	<u>Funciones de Restricción:</u>	<u>Funciones de Tablas de Resumen:</u>	<u>Funciones de Unión:</u>	<u>Funciones de Inteligencia de Tiempo:</u>
ALL	CALCULATETABLE	SUMMARIZE	EXCEPT	DATEADD
ALLNOBLANK	FILTER	SUMMARIZECOLUMNS	INTERSECT	DATESYTD
ALLEXCEPT	SUBSTITUTEWITHINDEX	GROUPBY	UNION	DATESQTD
ALLSELECTED	TOPN		ROW	DATESMTD
DISTINCT	TOPNPERLEVEL		GENERATE	DATESBETWEEN
CALENDARAUTO	TOPNSKIP		GENERATEALL	DATESINPERIOD
CALENDAR	<u>Funciones de Generación de Nuevas Columnas:</u>		NATURALINNERJOIN	FIRSTDATE
CROSSJOIN			NATURALLEFTOUTERJOIN	FIRSTNONBLANK
FILTERS	ADDCOLUMNS			LASTDATE
VALUES	SELECTCOLUMNS			LASTNONBLANK



# Clasificación de Funciones en el Lenguaje DAX

Según Resultado, Comportamiento y Proceso



## Según Resultado

Esta forma de clasificación de las funciones del lenguaje DAX consiste en poder identificar fácilmente si la función devuelve como resultado final un valor único o atómico, un conjunto de valores o tabla de datos, o absolutamente nada.

- **Tabular:** Función que devuelve un conjunto de valores en formato de tabla, es decir, es un arreglo de valores escalares.

### Funciones de Listas:

ALL  
ALLNOBLANK  
ALLEXCEPT  
ALLSELECTED  
DISTINCT  
CALENDARAUTO  
CALENDAR  
CROSSJOIN  
FILTERS  
VALUES

### Funciones de Restricción:

CALCULATETABLE  
FILTER  
SUBSTITUTEWITHINDEX  
TOPN  
TOPNPERLEVEL  
TOPNSKIP

### Funciones de Generación de Nuevas Columnas:

ADDCOLUMNS  
SELECTCOLUMNS

### Funciones de Tablas de Resumen:

SUMMARIZE  
SUMMARIZECOLUMNS  
GROUPBY

### Funciones de Unión:

EXCEPT  
INTERSECT  
UNION  
ROW  
GENERATE  
GENERATEALL  
NATURALINNERJOIN  
NATURALLEFTOUTERJOIN

### Funciones de Inteligencia de Tiempo:

DATEADD  
DATESYTD  
DATESQTD  
DATESMTD  
DATESBETWEEN  
DATESINPERIOD  
FIRSTDATE  
FIRSTNONBLANK  
LASTDATE  
LASTNONBLANK

# Clasificación de Funciones en el Lenguaje DAX

*Según Resultado, Comportamiento y Proceso*



## Según Resultado

Esta forma de clasificación de las funciones del lenguaje DAX consiste en poder identificar fácilmente si la función devuelve como resultado final un valor único o atómico, un conjunto de valores o tabla de datos, o absolutamente nada.

- **Modificadores:** Son funciones que devuelve nada, pues su objetivo es alterar el comportamiento de otras funciones para que cambien su comportamiento.

*Algunas funciones que actúan como modificadores:*

Para **CALCULATE** y **CALCULATETABLE** (Entre Otras)

**USERELATIONSHIP**

**REMOVEFILTERS**

**CROSSFILTER**

**ALLCROSSFILTERD** (Y todas la familia de funciones **ALLxxxx** para una de sus cara como modificador)

**KEEPFILTERS**

Para **SUMMARIZE** y/o **SUMMARIZECOLUMNS**

**ROLLUP**

**ROLLUPGROUP**

**ROLLUPADDSUBTOTAL**

**ROLLUPISSUBTOTAL**

# Clasificación de Funciones en el Lenguaje DAX

*Según Resultado, Comportamiento y Proceso*



## Según Comportamiento Respecto a Contextos

Esta forma de clasificación se refiere a cómo interactúa una función específica con los contextos de evaluación, dicho en otras palabras, si trabaja en el contexto de fila o si no, si trabaja en el contexto de filtro o sino.

### CONTEXTO DE FILA

- **Ignora el Contexto de Fila:** Trabaja por fuera del contexto de fila, en toda la tabla.
- **Tiene en Cuenta el Contexto de Fila:** Se calcula en la restricción de la fila actual

### CONTEXTO DE FILTRO

- **Ignora el Contexto de Filtro:** Trabaja en la tablas completas, tal cual fue carga al modelo de datos.
- **Tiene en Cuenta el Contexto de Filtro:** Se calculan en las tablas filtradas.

# Clasificación de Funciones en el Lenguaje DAX

*Según Resultado, Comportamiento y Proceso*



## Según Proceso

Esta forma de clasificación se refiere a cómo hace la función de manera interna para llegar al resultado final, en otros términos, es una clasificación de cómo hace DAX para llegar al valor que presenta en la interfaz.

- **Funciones de Cálculo Directo:** Simplemente realizar el cálculo, internamente no tiene que hacer nada especial,
- **Funciones de Escaneo:** Escanea para una operación exclusiva de la función
- **Funciones de Iteración:** Escanea la tabla y tenemos control de la operación en cada registro.
- **Funciones de Producto Cartesiano:** Toma dos o más listas elementos y realiza los cruces para generar todas las posibles combinaciones.

A conceptual image with a dark teal background. A hand from the bottom right reaches up, holding a glowing green infinity symbol. Behind the symbol is a faint bar chart with several bars of varying heights. The overall theme is iteration and continuous improvement.

# Funciones De Iteración

# Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*



**Funciones de Iteración:** Las funciones de iteración se ejecutan en una base fila a fila, es decir, inicia desde la primera y una a una va ejecutando un cálculo o requerimiento hasta el final, para con esos valores parciales ejecutar una gran operación que se corresponde con la semántica de la función

1

Funciones de Iteración Escalares

2

Funciones de Iteración Tabulares



## Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*

**=FUNCIONX ( Tabla ; Expresión )**

= **SUMX** ( Pedidos ; Pedidos[A] + Pedidos[B] - ( Pedidos[C] ^ 2 ) )

País	A	B	C
Colombia	1	2	3
Colombia	2	4	5
Colombia	2	2	1

$1 + 2 - 9 = -6$



## Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*

**=FUNCIONX ( Tabla ; Expresión )**

**= SUMX ( Pedidos ; Pedidos[A] + Pedidos[B] - ( Pedidos[C] ^ 2 ) )**

País	A	B	C
Colombia	1	2	3
Colombia	2	4	5
Colombia	2	2	1

$$1 + 2 - 9 = -6$$

$$2 + 4 - 25 = -19$$





## Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*

**=FUNCIONX ( Tabla ; Expresión )**

= **SUMX** ( Pedidos ; Pedidos[A] + Pedidos[B] - ( Pedidos[C] ^ 2 ) )

País	A	B	C
Colombia	1	2	3
Colombia	2	4	5
Colombia	2	2	1

$$1 + 2 - 9 = -6$$

$$2 + 4 - 25 = -19$$

$$2 + 2 - 1 = 3$$

$$\Sigma = -22$$

# Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*



**=FUNCIONX ( Tabla ; Expresión )**

**= SUMX (...)**

- 6  
- 19  
3

$\Sigma = - 22$

**= AVERAGEX (...)**

- 6  
- 19  
3

. = -7,3

**= MEDIANX (...)**

- 6  
- 19  
3

. = -6

**= MAXX (...)**

- 6  
- 19  
3

.á = 3

**= CONCATENATEX (...)**

- 6  
- 19  
3

= "-6,-19,3 "

## Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*

Funciones de iteración  
de Agregación  
0  
Funciones de Sufijo  
“X”

1	AVERAGEX		11	PRODUCTX	
2	CONCATENATEX		12	RANKX	
3	COUNTAX		13	STDEVX.P	
4	COUNTX		14	STDEVX.S	
5	GEOMEANX		15	SUMX	
6	MAXX		16	VARX.P	
7	MEDIANX		17	VARX.S	
8	MINX		18	XIRR	
9	PERCENTILX.EXC		19	XNPV	
10	PERCENTILX.INC				

Aunque este par de funcione devuelve un valor escalar, no tienen el sufijo X puesto que no hay que distinguirlas de otro par con los mismo nombres.



## Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*

**=FUNCIONX ( Tabla ; Expresión )**

**= SUMX ( Pedidos ; Pedidos[A] )**

País	A	B	C
Colombia	1	2	3
Colombia	2	4	5
Colombia	2	2	1

|



## Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*

**=FUNCIONX ( Tabla ; Expresión )**

**= SUMX ( Pedidos ; Pedidos[A] )**

País	A	B	C
Colombia	1	2	3
Colombia	2	4	5
Colombia	2	2	1

1

2



## Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*

**=FUNCIONX ( Tabla ; Expresión )**

= **SUMX** ( Pedidos ; Pedidos[A] )

País	A	B	C
Colombia	1	2	3
Colombia	2	4	5
Colombia	2	2	1

1  
2  
2

$\Sigma = 5$

# Funciones de Iteración de Agregación

*Funciones de sufijo X y su anatomía general*



## LA PRIMER SYNTAX SUGAR

Las funciones más simples equivalentes de las funciones de iteración escalares, realmente son una Syntax Sugar de las funciones de iteración.

**SUM** ( Pedidos[Ingresos] )

**SUMX** ( Pedidos ; Pedidos[Ingresos] )

**COUNTA** ( Pedidos[Ingresos] )

**COUNTAX** ( Pedidos ; Pedidos[Ingresos] )

**AVERAGE** ( Pedidos[Ingresos] )

**AVERAGEX** ( Pedidos ; Pedidos[Ingresos] )

# Funciones de Iteración de Agregación

*Funciones de Iteración de Tabla – Requieren Estudio Separado.*

## Funciones de iteración de Tabla

1	ADDCOLUMNS	
2	FILTER	
3	FIRSTNONBLANK	



Diciembre 2019

4

FIRSTNONBLANKVALUE



5

GENERATE



6

GENERATEALL






7

LASTNONBLANK



 Diciembre 2019

8	LASTNONBLANKVALUE	
9	SAMPLE	
10	SELECTCOLUMNS	

11	SUBSTITUWITINDEX	
	Evitar Operación de Generación	<a href="#">[Más aquí]</a>
12	SUMMARIZE	
13	TOPN	
14	TOPNSKIP	





# Funciones de Iteración de Agregación

Funciones de sufijo X y su anatomía general, detalles

## ANATOMÍA DE FUNCONES DE ITERACIÓN: BLOQUE DE CREACIÓN

Toda función de iteración tiene dos componentes como hemos estudiado:

- 1 **Tabla:** Aquí puede ser algo tan sencillo como llamar a una tabla existente en el modelo ejemplo: *Pedidos* o algo más complejo como expresión de tabla, ejemplo: *FILTER ( Pedidos ; Pedidos[País] = "Colombia" )*.
- 2 **Expresión:** Una expresión de tipo escalar de tipo numérico a ejecutarse fila a fila, eso si la expresión en la iteración no admite como resultados valores TRUE/FALSE e ignora valores null.

Ejemplo: `SUMMARIZE ( Pedidos ; Calendario[Fecha] ; "@ Ing" ; SUM ( Pedidos[Ingresos] ) )`

= **SUMX** ( Tabla ; Expresión )

Ejemplo: `ROUNDUP(Pedidos[Costo Producto]) + ROUNDUP(Pedidos[Costo Producto])`

# Métricas de Razones

## La Función ALL

*La Función ALL es una Función Tabular que Ignora el Contexto*



Hemos visto que la función **FILTER** nos permite restringir aún mas las filas disponibles, puesto que, en su proceso iterativo incluye al final sólo las que cumplan la condición de su segundo parámetro.

En ocasiones queremos hacer la operación contraria, **extender el número de filas disponibles** para que sea considerado en cierto cálculo.

Las funciones en el lenguaje DAX que cumplen este propósito son: las funciones **ALLxxx** y uno que otro alias de las mismas, sin embargo, la función más común y por excelencia para hacer esta operación es la función : **ALL**.

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1

Identificar Filtros

2

País	SKU	Ingresos	Utilidad
Argentina	L07	2	0,5
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

2

Aplicar Filtros

```
Tota_General :=  
SUMX ( Pedidos ;  
        Pedidos[Ingresos]  
)
```

3

Ejecutar Expresión DAX

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1

Identificar Filtros

	País	SKU	Ingresos	Utilidad
	Argentina	L07	2	0,5
2	Colombia	CB01	2	0,2
4	Colombia	L01	4	1
	Colombia	CB01	3	0,8

2

Aplicar Filtros

```
Tota_General :=
SUMX ( Pedidos ;
      Pedidos[Ingresos]
)
```

3

Ejecutar Expresión DAX

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1

Identificar Filtros

	País	SKU	Ingresos	Utilidad
	Argentina	L07	2	0,5
2	Colombia	CB01	2	0,2
4	Colombia	L01	4	1
3	Colombia	CB01	3	0,8

2

Aplicar Filtros

```
Tota_General :=  
SUMX ( Pedidos ;  
        Pedidos[Ingresos]  
)
```

3

Ejecutar Expresión DAX

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País	#VentasCB01
Argentina	
Brasil	
Chile	
Colombia	9
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

País	SKU	Ingresos	Utilidad
Argentina	L07	2	0,5
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

$\Sigma=9$

1

Identificar Filtros

2

Aplicar Filtros

3

Ejecutar Expresión DAX

```
Tota_General :=
SUMX ( Pedidos ;
      Pedidos[Ingresos]
)
```

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País	#VentasCB01
Argentina	
Brasil	
Chile	
Colombia	9
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

País	SKU	Ingresos	Utilidad
Argentina	L07	2	0,5
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

$\Sigma=9$

1

Identificar Filtros

2

Aplicar Filtros

3

Ejecutar Expresión DAX

```
Tota_General :=
SUMX ( Pedidos ;
      Pedidos[Ingresos]
)
```



# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1

Identificar Filtros

País	SKU	Ingresos	Utilidad
Argentina	L07	2	0,5
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

2

Aplicar Filtros

```
Tota_General :=  
SUMX ( Pedidos ;  
        Pedidos[Ingresos]  
)
```

3

Ejecutar Expresión DAX

## La Función ALL

*La Función ALL es una Función Tabular que Ignora el Contexto*



ALL

Tabular

*Una tabla completa o una tabla con una o más columnas*

XL ≥ 2010, PBI ≥ Nov 2016, SSAS ≥ 2012

La función **ALL** en su **primera cara** retorna la tabla especificada de forma material ignorando el contexto de filtro, es decir, no toma en cuenta ningún filtro que haya sido aplicado teniendo acceso a los pedacitos de tablas no visibles. Cuando se utiliza **ALL** con su **Primera Cara** tiene **dos variantes** en sus argumentos. **Variante Uno:** ALL en su primera variante retorna la tabla especificada de forma material ignorando el contexto de filtro, es decir, no toma en cuenta ningún filtro que haya sido aplicado. **Variante Dos:** Puede recibir uno o más parámetros, donde en cada uno se especifica una columna

- **SINTAXIS, VARIANTE I:**

ALL ( <Nombre de Tabla> )

- **SINTAXIS, VARIANTE II:**

ALL ( <Columna\_1> ; [Columna\_2] )

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

	País	SKU	Ingresos	Utilidad
2	Argentina	L07	2	0,5
	Colombia	CB01	2	0,2
	Colombia	L01	4	1
	Colombia	CB01	3	0,8

```
Tota_General :=  
SUMX ( ALL ( Pedidos ) ;  
Pedidos[Ingresos]  
)
```

1

Identificar Filtros

2

Aplicar Filtros

3

Ejecutar Expresión DAX

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1

Identificar Filtros

	País	SKU	Ingresos	Utilidad
2	Argentina	L07	2	0,5
2	Colombia	CB01	2	0,2
	Colombia	L01	4	1
	Colombia	CB01	3	0,8

2

Aplicar Filtros

```
Tota_General :=  
SUMX ( ALL ( Pedidos ) ;  
Pedidos[Ingresos]  
)
```

3

Ejecutar Expresión DAX

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1

Identificar Filtros

	País	SKU	Ingresos	Utilidad
2	Argentina	L07	2	0,5
2	Colombia	CB01	2	0,2
4	Colombia	L01	4	1
	Colombia	CB01	3	0,8

2

Aplicar Filtros

```
Tota_General :=  
SUMX ( ALL ( Pedidos ) ;  
      Pedidos[Ingresos]  
      )
```

3

Ejecutar Expresión DAX

# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País #VentasCB01

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1

Identificar Filtros

	País	SKU	Ingresos	Utilidad
2	Argentina	L07	2	0,5
2	Colombia	CB01	2	0,2
4	Colombia	L01	4	1
3	Colombia	CB01	3	0,8

2

Aplicar Filtros

```
Tota_General :=  
SUMX ( ALL ( Pedidos ) ;  
Pedidos[Ingresos]  
)
```

3

Ejecutar Expresión DAX



# La Función ALL

La Función ALL es una Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO

País  
Colombia

País	#VentasCB01
Argentina	
Brasil	
Chile	
Colombia	11
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

	País	SKU	Ingresos	Utilidad
2	Argentina	L07	2	0,5
2	Colombia	CB01	2	0,2
4	Colombia	L01	4	1
3	Colombia	CB01	3	0,8

$\Sigma=11$

1

Identificar Filtros

2

Aplicar Filtros

3

Ejecutar Expresión DAX

```
Tota_General :=  
SUMX ( ALL ( Pedidos ) ;  
        Pedidos[Ingresos]  
      )
```

## Aplicación de la Función ALL

Esta función es especialmente útil en una primera instancia para calcular el porcentaje de total general o la participación.

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	Σ=31



ParPais =

```

31 VAR IngresosPais =
    SUM ( Pedidos[Ingresos] )
VAR IngresosTodosLosPaises =
    SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )
RETURN
    DIVIDE ( IngresosPais; IngresosTodosLosPaises )

```

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX



# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	Σ=31



ParPais =

```

31 VAR IngresosPais =
    SUM ( Pedidos[Ingresos] )
VAR IngresosTodosLosPaises =
    SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )
RETURN
    DIVIDE ( IngresosPais; IngresosTodosLosPaises )

```

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País  
Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

2

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

2  
1

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País  
Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

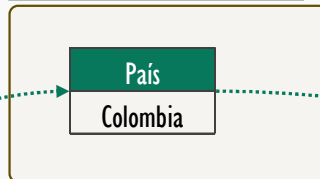
2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)



País	ParPais
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

1 Identificar Filtros

	SKU	País	Fecha	Ingresos
2	CB01	Argentina	1/04/2015	2
1	CB01	Argentina	1/06/2015	1
5	L07	Colombia	1/04/2015	5
2	L02	Colombia	4/04/2015	2
	L07	Colombia	12/05/2015	6
	CB01	Colombia	7/06/2015	4
	CB01	Colombia	7/09/2015	3
	L01	Colombia	8/11/2015	2
	CB01	Colombia	9/11/2015	9
	L07	Brasil	9/11/2015	7

2 Aplicar Filtros

```

ParPais =
31 VAR IngresosPais =
    SUM ( Pedidos[Ingresos] )
VAR IngresosTodosLosPaises =
    SUMX ( ALL ( Pedidos ), Pedidos[Ingresos] )
RETURN
    DIVIDE ( IngresosPais; IngresosTodosLosPaises )

```

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

	SKU	País	Fecha	Ingresos
2	CB01	Argentina	1/04/2015	2
1	CB01	Argentina	1/06/2015	1
5	L07	Colombia	1/04/2015	5
2	L02	Colombia	4/04/2015	2
6	L07	Colombia	12/05/2015	6
4	CB01	Colombia	7/06/2015	4
	CB01	Colombia	7/09/2015	3
	L01	Colombia	8/11/2015	2
	CB01	Colombia	9/11/2015	9
	L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

	SKU	País	Fecha	Ingresos
2	CB01	Argentina	1/04/2015	2
1	CB01	Argentina	1/06/2015	1
5	L07	Colombia	1/04/2015	5
2	L02	Colombia	4/04/2015	2
6	L07	Colombia	12/05/2015	6
4	CB01	Colombia	7/06/2015	4
3	CB01	Colombia	7/09/2015	3
	L01	Colombia	8/11/2015	2
	CB01	Colombia	9/11/2015	9
	L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX



# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País  
Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

1 Identificar Filtros

	SKU	País	Fecha	Ingresos
2	CB01	Argentina	1/04/2015	2
1	CB01	Argentina	1/06/2015	1
5	L07	Colombia	1/04/2015	5
2	L02	Colombia	4/04/2015	2
6	L07	Colombia	12/05/2015	6
4	CB01	Colombia	7/06/2015	4
3	CB01	Colombia	7/09/2015	3
2	L01	Colombia	8/11/2015	2
9	CB01	Colombia	9/11/2015	9
	L07	Brasil	9/11/2015	7

2 Aplicar Filtros



```

ParPais =
31 VAR IngresosPais =
    SUM ( Pedidos[Ingresos] )
VAR IngresosTodosLosPaises =
    SUMX ( ALL ( Pedidos ), Pedidos[Ingresos] )
RETURN
    DIVIDE ( IngresosPais; IngresosTodosLosPaises )
  
```

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7



ParPais =

31 VAR IngresosPais =

SUM ( Pedidos[Ingresos] )

VAR IngresosTodosLosPaises =

SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )

RETURN

DIVIDE ( IngresosPais; IngresosTodosLosPaises )

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7

$\Sigma=41$



ParPais =

```

31 VAR IngresosPais =
    SUM ( Pedidos[Ingresos] )
41 VAR IngresosTodosLosPaises =
    SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )
RETURN
    DIVIDE ( IngresosPais; IngresosTodosLosPaises )

```

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7

$\Sigma=41$



ParPais =

```

31 VAR IngresosPais =
    SUM ( Pedidos[Ingresos] )
41 VAR IngresosTodosLosPaises =
    SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )
RETURN
    DIVIDE ( IngresosPais; IngresosTodosLosPaises )

```

1 Identificar Filtros

2 Aplicar Filtros

3 Ejecutar Expresión DAX

# La Función ALL

La Función ALL es faceta como Función Tabular que Ignora el Contexto

CONTEXTO DE FILTRO (ORIGINAL)

País

Colombia

País ParPais

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

0,75

1 Identificar Filtros

SKU	País	Fecha	Ingresos
CB01	Argentina	1/04/2015	2
CB01	Argentina	1/06/2015	1
L07	Colombia	1/04/2015	5
L02	Colombia	4/04/2015	2
L07	Colombia	12/05/2015	6
CB01	Colombia	7/06/2015	4
CB01	Colombia	7/09/2015	3
L01	Colombia	8/11/2015	2
CB01	Colombia	9/11/2015	9
L07	Brasil	9/11/2015	7

2 Aplicar Filtros



ParPais =

```

31 VAR IngresosPais =
    SUM ( Pedidos[Ingresos] )
41 VAR IngresosTodosLosPaises =
    SUMX ( ALL ( Pedidos ); Pedidos[Ingresos] )
RETURN
    DIVIDE ( IngresosPais; IngresosTodosLosPaises )

```

31  
41

$\frac{31}{41} = 0,75$

3 Ejecutar Expresión DAX



## Evaluando Tu Entendimiento

*Funciones Tabulares de Primer Nivel (Libres) en Medidas*



### ¿La Función ALL y la Función FILTER?

*Modifican el Contexto de Filtro*



## Evaluando Tu Entendimiento

*Contexto de Filtro y Contexto de Fila*



*Pregunta*



**Puedes Decir el Resultado de la  
Siguiete Medida**

***=SUMX ( Pedidos ;  
COUNTROWS ( Pedidos )  
)***



## Análisis:

## Evaluando Tu Comprensión 2

=SUMX ( Pedidos ; COUNTROWS (Pedidos) )

## CONTEXTO DE FILTRO

País  
Colombia

País	#VtsAl2
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

1

Identificar Filtros

País	SKU	Ingresos	Utilidad
Argentina	L07	2	0,5
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

2

Aplicar Filtros

```
= SUMX ( Pedidos ;  
          COUNTROWS (Pedidos)  
        )
```

País	SKU	Ingresos	Utilidad
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

3

Ejecutar Expresión DAX

## Análisis:

## Evaluando Tu Comprensión 2

=SUMX ( Pedidos ; COUNTROWS ( Pedidos ) )

## CONTEXTO DE FILTRO

País  
Colombia

País	#VtsAl2
Argentina	
Brasil	
Chile	
Colombia	
Ecuador	
Paraguay	
Perú	
Uruguay	
Venezuela	

1

Identificar Filtros

País	SKU	Ingresos	Utilidad
Argentina	L07	2	0,5
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

2

Aplicar Filtros

```
= SUMX ( Pedidos ;  
          COUNTROWS ( Pedidos )  
        )
```

País	SKU	Ingresos	Utilidad
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

3

Ejecutar Expresión DAX

## Análisis:

## Evaluando Tu Comprensión 2

$$=SUMX ( Pedidos ; COUNTROWS ( Pedidos ) )$$

## CONTEXTO DE FILTRO

País  
Colombia

País #VtsAl2

Argentina  
Brasil  
Chile  
Colombia  
Ecuador  
Paraguay  
Perú  
Uruguay  
Venezuela

9

1

Identificar Filtros

País	SKU	Ingresos	Utilidad
Argentina	L07	2	0,5
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

2

Aplicar Filtros

$$=SUMX ( Pedidos ; COUNTROWS ( Pedidos ) )$$

País	SKU	Ingresos	Utilidad
Colombia	CB01	2	0,2
Colombia	L01	4	1
Colombia	CB01	3	0,8

3

3

 $\Sigma = 9$ 

3

Ejecutar Expresión DAX

# Siguiente Paso:

Round 04, Jueves:

## Cálculos Tipo Excel, Celda Siguiente y Celda Anterior

- Leguaje DAX de Cero a Guerrero  
**[DAX con Power BI]**

[www.excelfreeblog.com](http://www.excelfreeblog.com)

[www.escueladeinteligenciadenegocios.com](http://www.escueladeinteligenciadenegocios.com)



Escuela de Inteligencia de Negocios | [ExcelFreeBlog.com](http://ExcelFreeBlog.com)