



Caso 402 Distribuciones Canal HORECA

04 Casos Avanzando con DAX. Aplicando la función RANKX

Jose Ignacio González Gómez

Departamento de Economía, Contabilidad y Finanzas - Universidad de La Laguna

www.jggomez.eu

V.2.3

Ejercicio Basado: [Todo Sobre RANKX en POWER BI](#)

Archivos fuentes: Power BI básico con modelo de datos y medidas basicas asi como informe base ya diseñado, solo trabajar con las medidas solicitadas para hacer el ranking.

Funciones DAX tratadas: SUM, SUMX, DIVIDE, IF, HASONEVALUE, ALL, ALLSELECTED, RANKX, VAR, BLANK

Funciones de inteligencia de tiempo sameperiodlastyear

Elementos visuales aplicados: Tabla básicas

Contenido

Presentacion del caso.....	2
Información disponible	2
El informe base y objetivo.....	2
Se pide.....	3
Solución	4
RankingGlobal y RankingSeleccion.....	4
Con ALL.....	4
Con ALLSELECTED	4
RankingGlobal y RankingSeleccion depurado sin vacíos ni negativos	5
Justificación	5
Con ALL e IF, depurando para vacíos y no negativos	5
Con ALLSELECTED e IF, depurando para vacíos y no negativos	5
Quitando los totales con HASONEVALUE.....	6
Complementando con inteligencia de tiempo, sameperiodlastyear	7
Anexo	8
Propuesta de creacion de tabla calendario.....	8
Fundamentos técnicos, DAX - RANKX	8
Sintaxis y parámetros	8
Ámbito de aplicación. RANKX con ALL, ALLEXCEPT, ALLNOBLANKROW y ALLSELECTED.....	8
Ejemplo.....	9

Presentacion del caso

Información disponible

Disponemos de un modelo de datos ya elaborado que contienen las ventas por producto durante tres años y las medidas basicas correspondiente

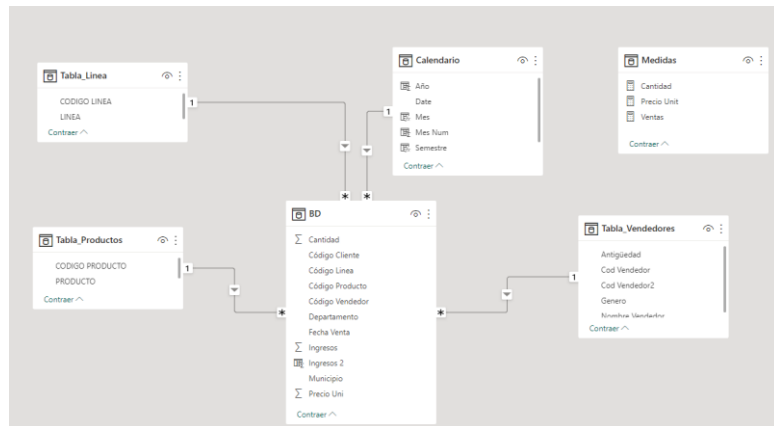
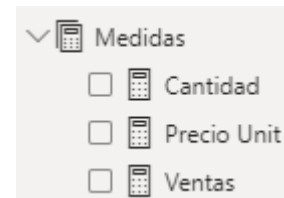


Ilustración 1

Las medidas previamente creadas y que disponemos en el modelo son las siguientes que debemos analizar.

Las medidas previamente creadas y que disponemos en el modelo son las siguientes que debemos analizar.

- **Cantidad = SUM(BD[Cantidad])**
- **Precio Unit = DIVIDE([Ventas], [Cantidad])**
- **Ventas = SUMX(BD, BD[Cantidad]*BD[Precio Uni])**



El informe base y objetivo

Nuestro informe base es el siguiente, tenemos de una tabla que contiene los productos y una medida de ventas que es la suma del producto de precio unitario por cantidad. En la ilustración mostramos la medida condicional a las ventas de cada producto para el 2021

Año	PRODUCTO	Ventas
2019	ACEITU VERD MAN HUESO S E N420-D250	201.760
2020	AGUA PET MONTECLARO CRISTAL 600 mL	2.064.960
2021	ALPISTE X 500G	2.489.192
	ARVEJA X 500G	4.188.340
	AVENA EN HARINA SABOR FRESA SAN ESTEBAN BOLSA 200 G	26.121.960
	AVENA EN HARINA SABOR VAINILLA	60.516.000
	AVENA EN HOJUELA	65.265.246
	AVENA EN HOJUELA BANANO EN TROZOS	12.205.116
	AVENA EN HOJUELA CHOCOLATE	44.634.071
	AVENA EN HOJUELA CORTADA	41.363.667
	AVENA EN HOJUELA-FRESA TAEQ	45.567.235
	AVENA EN HOJUELA GRANOLERA	47.516.400
	AVENA EN HOJUELA MASMELOS	30.098.745
	AZUCAR MOLIDA	57.926.917
	AZUCAR MORENA	55.124.927
	CAJA CEREALES 49.5 CM X 33.5 CM X 24.0 CM	48.038.253
	CALDO GALLINA x 1.8kgr x 165 CUBOS	22.136.029
	CALDO GALLINA x 2.97 kgr X 270 CUBOS	51.678.708
	CALDO RICOSTILLA CUBOS x 60x660gr	18.179.648
	Total	16.043.777.262

Ilustración 2

Nuestro objetivo es disponer de una medida que nos asigne el ranking de ventas de los productos de manera consecutiva condicionado al contexto filtro afectado (Año – Tipo de Producto - Vendedor). Esta función es RANKX que es la que vamos a estudiar.

Analizar el ranking que ocupa cada producto de la tabla para el periodo seleccionado. Medida RankxTotal

Se pide

Realizar un informe simple basado en una sola tabla, varios segmentadores y con las siguientes medidas.

PRODUCTO	Ventas	RankingGlobal	RankingSeleccion	RankingGlobalLY	RankingSeleccionLY
ALPISTE X 500G	2.489.192	112	1		
AGUA PET MONTECLARO CRISTAL 600 mL	2.064.960	114	2		
AGUA GAS PET MONTECLARO 600 mL				116	1
ACEITUNA RELL PIMIENTO M S ESP N420-D235				126	2
ACEITU VERD MAN HUESO S E N420-D250	201.760	132	3	135	3
Total	4.755.912				

Ilustración 3

- **RankingGlobal.** Ranking basado en la medida “Ventas” (total ventas) de cada producto globalmente, es decir para todos los productos vendidos en el periodo seleccionado, independientemente de la selección de productos. **Empleo de la función ALL.** En general sería algo similar a lo siguiente:

RankingGlobal =

`RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)`

- **RankingSelección.** Ranking basado en la medida “Ventas” (total ventas) de cada producto pero teniendo en cuenta el contexto de filtro, es decir el ranking se establece para aquellos productos seleccionados en el segmentador, es decir para todos los productos vendidos en el periodo seleccionado y considerando los elementos definidos en el segmentador. **Empleo de la función ALLSELECTED.** En general sería algo similar a lo siguiente:

RankingSeleccion =

`RANKX(ALLSELECTED(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)`

Y para las medidas de **inteligencia de tiempo** tendríamos las siguientes:

RankingGlobalLY =

`CALCULATE([RankingGlobal],SAMEPERIODLASTYEAR((Calendario[Date])))`

RankingSeleccionLY =

`CALCULATE([RankingSeleccion],SAMEPERIODLASTYEAR((Calendario[Date])))`

Solución

RankingGlobal y RankingSeleccion

Con ALL

Queremos establecer un ranking general independientemente del contexto, es decir el ranking que ocupa cada elemento evaluado sin tener en cuenta el contexto de filtro que este afectando, por ejemplo vamos a analizar qué productos son los más vendidos -independientemente del contexto- y en qué posición se ordenan tras este análisis. En este caso la formula será combinar la función RANKX con ALL.

RankingGlobal = RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)

El resultado es un ranking basado en toda la tabla de productos y por tanto omite el contexto de filtro, tal y como podemos observar en la siguiente ilustración. Si seleccionamos solo 6 productos en el segmentador, el ranking me lo establece para el conjunto global de los todos los productos seleccionados y no seleccionados, omitiendo por tanto el contexto filtro.

PRODUCTO	Ventas	RankingGlobal
ALPISTE X 500G	2.489.192	112
AGUA PET MONTECLARO CRISTAL 600 mL	2.064.960	114
ACEITU VERD MAN HUESO S E N420-D250	201.760	132
ACEITUN RELL ANCHOA M S ESP N350-D150		155
ACEITUNA RELL PIMIENTO M S ESP N420-D235		155
AGUA GAS PET MONTECLARO 600 mL		155
Total	4.755.912	3 104

RankingGlobal = RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)

Ilustración 4

Con ALLSELECTED

Nos interesa este ranking para aquellos productos seleccionado según el segmentador y por tanto que tenga en cuenta el contexto filtro que afecta a la tabla y para ello debemos usar la función ALLSELECTED.

RankingSeleccion = RANKX(ALLSELECTED(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)

PRODUCTO	Ventas	RankingGlobal	RankingSeleccion
ACEITU VERD MAN HUESO S E N420-D250	201.760	132	3
ACEITUN RELL ANCHOA M S ESP N350-D150		155	4
ACEITUNA RELL PIMIENTO M S ESP N420-D235		155	4
AGUA GAS PET MONTECLARO 600 mL		155	4
AGUA PET MONTECLARO CRISTAL 600 mL	2.064.960	114	2
ALPISTE X 500G	2.489.192	112	1
Total	4.755.912	3 104	3 1

RankingGlobal = RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
 RankingSeleccion = RANKX(ALLSELECTED(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)

Ilustración 5

Observamos en la ilustración anterior que disponemos de dos Ranking el Global o posición global del producto en la tabla general y el Ranking Selección de los productos o elementos seleccionados en el segmentador.

Así con esta función recorreremos toda la matriz de producto, pero teniendo en cuenta el contexto de filtro externo, es decir de selección de productos.

RankingGlobal y RankingSeleccion depurado sin vacíos ni negativos

Justificación

Pero nos puede interesar que los valores vacíos ni los negativos (devoluciones) no se evalúen para establecer el ranking (nota 1 de la Ilustración 5), es decir se excluya los productos no vendidos (valores vacíos o con 0) y los devueltos (aquellos que tienen valores en negativos) para lo cual adaptaremos las fórmulas anteriores con una condición utilizando la función IF y para facilitar la lectura del código hemos creado una variable que llamamos TablaRanking de tal forma que si la medida ventas es mayor que 0 (es decir no está vacía y es positiva) esta formara parte de la base para aplicar la fórmula RANKX y con RETURN devuelve la tabla generada, es decir:

```
Var TablaRanking
.....
RETURN
If ([Ventas]>0, TablaRanking)
```

Por tanto, las dos fórmulas modificadas quedarían así:

Con ALL e IF, depurando para vacíos y no negativos

RankingGlobal con ALL, para todos tendríamos

```
RankingGlobal = VAR TablaRanking=
RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
RETURN
If ([Ventas]>0, TablaRanking)
```

Con ALLSELECTED e IF, depurando para vacíos y no negativos

Y la otra sería:

```
RankingSeleccion = VAR TablaRanking=
RANKX(ALLSELECTED(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
RETURN
If ([Ventas]>0, TablaRanking)
```

Por tanto, el resultado final será:

PRODUCTO	Ventas	RankingGlobal	RankingSeleccion
ACEITU VERD MAN HUESO S E N420-D250	201.760	132	3
AGUA PET MONTECLARO CRISTAL 600 mL	2.064.960	114	2
ALPISTE X 500G	2.489.192	112	1
Total	4.755.912	104	1

```
RankingGlobal = VAR TablaRanking=
RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
RETURN
If ([Ventas]>0, TablaRanking)

RankingSeleccion = VAR TablaRanking=
RANKX(ALLSELECTED(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
RETURN
If ([Ventas]>0, TablaRanking)
```

Quitar el valor con la función HASONEVALUE

Ilustración 6

Como podemos observar (Ilustración 6) ahora se omiten para establecer el ranking global y el de selección los vacíos y los negativos

Quitando los totales con HASONEVALUE

Solo nos queda quitar los totales de la columna de ranking en cuanto que carece de sentido y para ello utilizaremos la función **HASONEVALUE** (ver Ilustración 6), que viene a decir que si no hay contexto, en nuestro caso nombre de producto, no aplique la función.

Así esta última adaptación para las dos fórmulas quedara de la siguiente forma incorporando:

```
IF(HASONEVALUE(Tabla_Productos[PRODUCTO])=TRUE,
.....
BLANK())
```

RankingGlobal con ALL, quitando vacíos y negativos y sin total

```
1 RankingGlobal =
2 IF(HASONEVALUE(Tabla_Productos[PRODUCTO])=TRUE,
3   VAR TablaRanking=
4     RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
5     RETURN
6     If ([Ventas]>0, TablaRanking),
7     BLANK())
```

```
RankingGlobal =
IF(HASONEVALUE(Tabla_Productos[PRODUCTO])=TRUE,
VAR TablaRanking=
RANKX(ALL(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
RETURN
If ([Ventas]>0, TablaRanking),
BLANK())
```

RankingSeleccion con ALLSELECTED, quitando vacíos y negativos y sin total

```
1 RankingSeleccion =
2 IF(HASONEVALUE(Tabla_Productos[PRODUCTO])=TRUE,
3   VAR TablaRanking=
4     RANKX(ALLSELECTED(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
5     RETURN
6     If ([Ventas]>0, TablaRanking),
7     BLANK())
```

```
RankingSeleccion =
IF(HASONEVALUE(Tabla_Productos[PRODUCTO])=TRUE,
VAR TablaRanking=
RANKX(ALLSELECTED(Tabla_Productos[PRODUCTO]),[Ventas],,DESC,Dense)
RETURN
If ([Ventas]>0, TablaRanking),
BLANK())
```

Este será el resultado con las formulas aplicadas.

PRODUCTO	Ventas	RankingGlobal	RankingSeleccion
ACEITU VERD MAN HUESO S E N420-D250	201.760	132	3
AGUA PET MONTECLARO CRISTAL 600 mL	2.064.960	114	2
ALPISTE X 500G	2.489.192	112	1
Total	4.755.912		

Ilustración 7

Se recomienda ver el anexo “¡Error! No se encuentra el origen de la referencia.”

Complementando con inteligencia de tiempo, sameperiodlastyear

Ademas deseamos disponer de una media que nos permita comparar el ranking tanto global como selectivo respecto al mismo periodo del ejercicio anterior y para ello es necesario trabajar con las medidas de inteligencia de tiempo, en nuestro caso concreto con sameperiodlastyear

Asi las dos medidas propuestas serán y que añadidas al modelo generan el siguiente resultado:

- RankingGlobalLY =
CALCULATE([RankingGlobal],SAMEPERIODLASTYEAR((Calendario[Date])))
- RankingSeleccionLY =
CALCULATE([RankingSeleccion],SAMEPERIODLASTYEAR((Calendario[Date])))

PRODUCTO	Ventas	RankingGlobal	RankingGlobalLY	RankingSeleccion	RankingSeleccionLY
ALPISTE X 500G	2.489.192	112		1	
AGUA PET MONTECLARO CRISTAL 600 mL	2.064.960	114		2	
AGUA GAS PET MONTECLARO 600 mL			116		1
ACEITUNA RELL PIMIENTO M S ESP N420-D235			126		2
ACEITU VERD MAN HUESO S E N420-D250	201.760	132	135	3	3
Total	4.755.912				

Ilustración 8

Anexo

Propuesta de creacion de tabla calendario

```

Calendario =
ADDCOLUMNS (
CALENDARAUOTO(),
"Año", YEAR ( [Date] ),
"Mes", FORMAT ( [Date], "MM" ),
"Periodo", FORMAT ( [Date], "YYYY/MM" ),
"Periodo2", FORMAT ( [Date], "YYYY/mmm" ),
"Nombre Corto Mes", FORMAT ( [Date], "mmm" ),
"Nombre Mes", FORMAT ( [Date], "mmm" ),
"Día Semana número", WEEKDAY ( [Date] ),
"Día Semana ", FORMAT ( [Date], "ddd" ),
"Día Semana Corto", FORMAT ( [Date], "ddd" ),
"Trimestre", "Q" & FORMAT ( [Date], "Q" ),
"Año/trimestre", FORMAT ( [Date], "YYYY" ) & "/" & FORMAT ( [Date], "Q" )
)
    
```

Fundamentos técnicos, DAX - RANKX

Sintaxis y parámetros

La función **RANKX** devuelve la posición o rango del cada valor resultante de aplicar una expresión a las filas de una tabla comparándolos entre sí. La función RANKX se compone de 5 argumentos de los cuales tres son opcionales:

Sintaxis:	Parámetros
RANKX (
table,	table : Tabla conteniendo las filas para las que se evaluará la expresión.
expression	expression : Expresión a evaluar para cada fila de la tabla
[, value	value : Argumento opcional . Escalar o expresión que devuelve un escalar que se desea buscar en la lista de valores a analizar
[, order	order : Argumento opcional . Valor que especifica cómo aplicar un rango a un valor (ordenando la lista de abajo arriba o de arriba abajo).
[, ties]]]	ties : Argumento opcional . Argumento que determina cómo aplicar rangos cuando hay igualdades en la lista
)	

Ver mas detalle de la función en: [RANKX | Interactive Chaos](#)

Ámbito de aplicación. RANKX con ALL, ALLEXCEPT, ALLNOBLANKROW y ALLSELECTED

Asi tenemos:

```

1 Ranking =
2 RANKX(ALL
3 | ALLSE fx ALL
4 [Ventas], fx ALLEXCEPT
5 DESC, fx ALLNOBLANKROW
6 Dense] fx ALLSELECTED
    
```

- **ALL** devuelve todas las filas de una tabla o todos los valores de una o varias columnas, ignorando cualquier filtro que se haya aplicado
- **AllExcept** La función **ALLEXCEPT** elimina todos los filtros de contexto aplicados a la tabla indicada salvo los que hagan referencia a las columnas incluidas como argumentos
- La función **ALLNOBLANKROW** recibe como argumento el nombre de una o varias columnas, o una tabla, y devuelve una tabla conteniendo las filas de la tabla, columna o

columnas indicadas sin incluir las filas formadas por valores BLANK e ignorando los posibles filtros que pudiesen estar aplicándose a la tabla, columna o columnas

Ejemplo

Supongamos que tenemos un listado de ventas en el que se incluye el producto vendido y el número de unidades vendidas. Hemos creado la medida.

Ventas totales = SUM(Ventas[Cantidad])

Y creamos una visualización de tabla

Producto	Cantidad
A	1
B	5
C	3
B	2
A	2
D	2
C	3

Producto	Ventas totales
A	3
B	7
C	6
D	2
Total	18

Deseásemos saber qué productos son los más vendidos -independientemente del contexto- y en qué posición se ordenan tras este análisis, la función a aplicar será RANKX con la siguiente sintaxis: **Rango = RANKX(ALL(Ventas[Producto]), Ventas[Ventas totales])**

Producto	Ventas totales	Rango
A	3	3
B	7	1
C	6	2
D	2	4
Total	18	1

Si añadimos esta medida a la tabla anterior obtenemos:

Dando el valor 1 al producto más vendido (el producto B, con 7 unidades) y el valor 4 al producto menos vendido (el producto D, con 2 unidades).

Obsérvese **la necesidad de incluir en la definición de la medida Rango la función ALL**. De otra forma, cuando la máquina DAX comenzase con el primero de los productos (el producto A), el contexto quedaría limitado a este producto. Es decir, la medida Rango se calcularía en este contexto limitado a las filas correspondientes al producto A. Y el orden que ocupan las ventas del producto A en una lista en la que solo se encuentra este producto es el 1 (es el producto más vendido pues es el único). Y lo mismo ocurriría con los demás productos: el rango se calcularía siempre considerando una tabla en la que solo se encontrase dicho producto, por lo que su rango sería siempre 1.

Supongamos ahora que deseamos que los rangos se apliquen en orden inverso, es decir, que se dé el valor 1 al producto menos vendido y el 4 al más vendido. Para esto bastaría con incluir el parámetro order:

Rango = RANKX(ALL(Ventas[Producto]), Ventas[Ventas totales], ,1)

Producto	Ventas totales	Rango
A	3	2
B	7	4
C	6	3
D	2	1
Total	18	5