Power BI

Trucos y buenas prácticas

Vol. IV



CONTENIDO

1. CONEXIÓN Y TRANSFORMACIÓN DE DATOS 3

1. COMBINAR FICHEROS EXCEL ADJUNTOS EN CORREOS ELECTRÓNICOS 3
2. COMBINAR FICHEROS EXCEL CON ENCABEZADOS DIFERENTES 8
3. ALTERNATIVA A LA FUNCIÓN "TRANSPONER" 14
4. USO DE REGISTROS COMO ALTERNATIVA A CREAR COLUMNAS PERSONALIZADAS 16
5. DIVIDIR CELDAS CON VARIAS LÍNEAS DE DATOS EN DIFERENTES REGISTROS 20
6. IGNORAR CASE SENSITIVITY 25
7. EXPANDIR LAS COLUMNAS SIN ESPECIFICAR EL NOMBRE DE ESTAS 27
8. DESAPILAR FILAS EN COLUMNAS 29

2. MODELADO 35

9. CONVERTIR MATRIZ EN DATOS TABULARES 35
10. FILTRAR VALORES ENTRE SEGMENTACIONES 39
11. RANGO ESTÁTICO DE LOS EJES 41
12. CALCULAR PORCENTAJES PRECISOS CON RLS 46

3. VISUALIZACIÓN 50

13. FORMATO CONDICIONAL BASADO EN CAMPOS DE TIPO STRING 50
14. ETIQUETAS DE DATOS PERSONALIZADAS CON MEDIDAS 52
15. USOS DEL FORMATO DINÁMICO 55
16. BARRAS DE DATOS EN TABLAS Y COLUMNAS 57
17. RESALTAR DATOS DINÁMICAMENTE SEGÚN LAS SEGMENTACIONES 60
18. RESALTAR VALORES MÍNIMO Y MÁXIMO DE UN GRÁFICO 64
19. PERSONALIZACIÓN DE SEGMENTACIONES 67

4. OTROS 73

20. TIPS&TRICKS SOBRE JERARQUÍAS EN SCORECARDS 73
21. LEER ROLES DE SEGURIDAD ACTIVOS EN POWER BI 75
22. ALTERNATIVA A SORT BY COLUMN 79
23. REPLICAR FUNCIONES SUMIF Y COUNTIF EN POWER BI 81
24. ALTERNATIVA DEL SERVICIO DE POWER BI A LAS TABLAS PIVOTANTES DE EXCEL 83

5. MAS TRUCOS, VIDEOTUTORIALES Y PAPERS 88

1. CONEXIÓN Y TRANSFORMACIÓN DE DATOS

1. Combinar ficheros Excel adjuntos en correos electrónicos

El propósito de este tip es mostrar cómo combinar ficheros de Excel recibidos por correo electrónico de forma periódica.



Desarrollo: Imaginemos que todos los meses se recibe un correo electrónico con la misma estructura del asunto del correo y un fichero Excel adjunto (que también mantiene la misma estructura) y queremos unificar todos los ficheros:

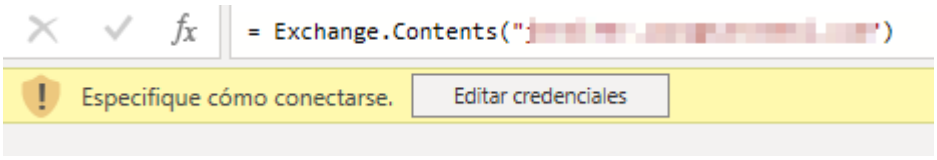
OrderDate	SalesOrderNumber	SalesAmount
01/01/2023	SO61294	4,99
02/01/2023	SO61302	2294,99
03/01/2023	SO61269	34,99
04/01/2023	SO61280	34,99
05/01/2023	SO61282	2,29
06/01/2023	SO61284	21,49
07/01/2023	SO61299	35
08/01/2023	SO61290	9,99
09/01/2023	SO61292	9,99
10/01/2023	SO61300	21,98
11/01/2023	SO61284	2,29
12/01/2023	SO61300	9,99
13/01/2023	SO61294	7,95
14/01/2023	SO61269	49,99
15/01/2023	SO61299	4,99
16/01/2023	SO61290	4,99
17/01/2023	SO61292	4,99
18/01/2023	SO61302	4,99
19/01/2023	SO61300	4,99
20/01/2023	SO61302	9,99

January

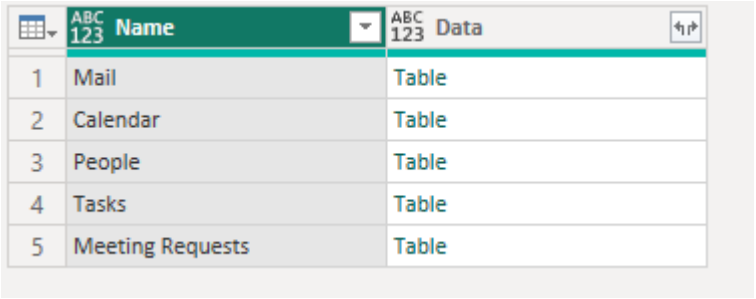
OrderDate	SalesOrderNumber	SalesAmount
01/02/2023	SO63316	4,99
02/02/2023	SO63317	4,99
03/02/2023	SO63313	4,99
04/02/2023	SO63355	539,99
05/02/2023	SO63352	2384,07
06/02/2023	SO63342	2294,99
07/02/2023	SO63335	769,49
08/02/2023	SO63319	21,98
09/02/2023	SO63306	28,99
10/02/2023	SO63316	120
11/02/2023	SO63317	54,99
12/02/2023	SO63319	54,99
13/02/2023	SO63313	8,99
14/02/2023	SO63352	8,99
15/02/2023	SO63355	8,99
16/02/2023	SO63335	34,99
17/02/2023	SO63306	4,99
18/02/2023	SO63342	4,99
19/02/2023	SO63352	4,99
20/02/2023	SO63355	4,99

February

Lo primero que haremos es abrir un pbix nuevo utilizando como origen de datos un fichero Excel en blanco y abrir Power Query para conectarnos a la bandeja de entrada del correo. Para ello, utilizamos la función Exchange.Contents y una cuenta de correo Microsoft:

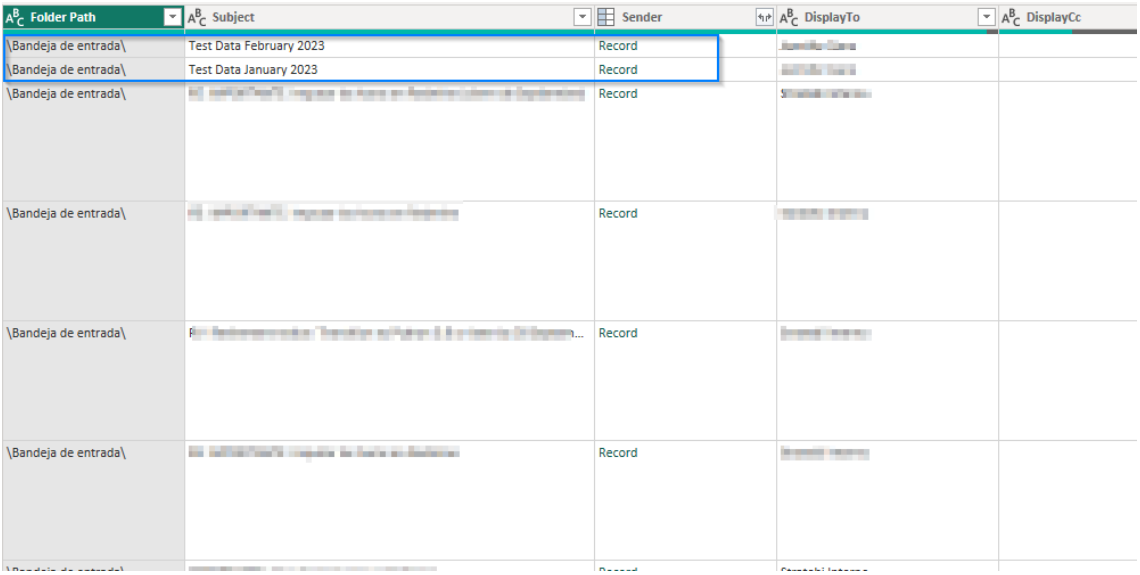


Introducimos las credenciales y tenemos conectado Power Query a nuestro correo electrónico:



	Name	Data
1	Mail	Table
2	Calendar	Table
3	People	Table
4	Tasks	Table
5	Meeting Requests	Table

El siguiente paso es expandir la tabla de la categoría “Mail”, pues es la única que nos interesa para este ejemplo:



Folder Path	Subject	Sender	DisplayTo	DisplayCc
\Bandeja de entrada\	Test Data February 2023	Record		
\Bandeja de entrada\	Test Data January 2023	Record		
\Bandeja de entrada\		Record		
\Bandeja de entrada\		Record		
\Bandeja de entrada\		Record		
\Bandeja de entrada\		Record		

Como podemos observar, aparecen todos los correos electrónicos de todas las categorías, por tanto, el siguiente paso es filtrar por el campo “Folder Path” por Bandeja de entrada:



A continuación, filtramos por el campo “Subject” de forma que solo nos quedemos con aquellos correos cuyo asunto comience por “Test Data”:

```
= Table.SelectRows(#"Filas filtradas", each Text.StartsWith([Subject], "Test Data"))
```

	Folder Path	Subject	Sender
1	\Bandeja de entrada\	Test Data February 2023	Record
2	\Bandeja de entrada\	Test Data January 2023	Record

Para evitar perder aquellos registros que tengan mayúsculas o minúsculas diferentes, añadimos Comparer.OrdinalIgnoreCase como parámetro en la función para deshabilitar el case sensitive:

```
= Table.SelectRows(#"Filas filtradas", each Text.StartsWith([Subject], "Test Data", Comparer.OrdinalIgnoreCase))
```

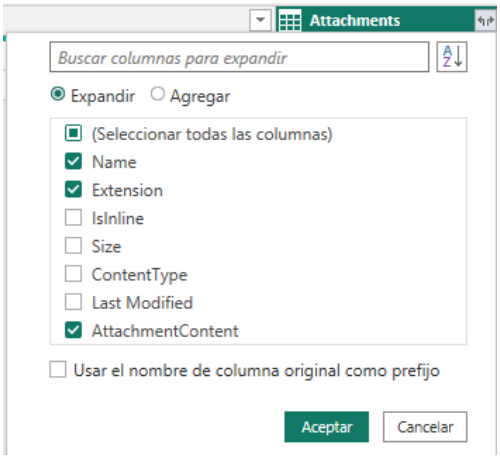
Además, vamos a filtrar por el campo “HasAttachments” = True.

HasAttachments
TRUE
TRUE

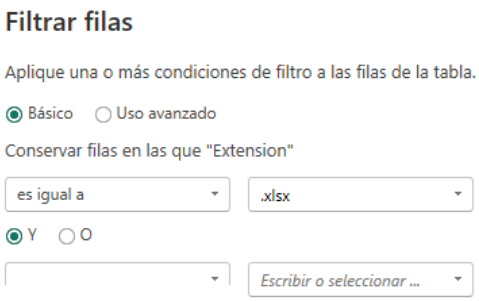
A continuación, seleccionamos únicamente las columnas que nos interesan (Subject y Attachments) y eliminamos las demás:

Subject	Attachments
Test Data February 2023	Table
Test Data January 2023	Table

Expandimos la columna de Attachments:



Filtramos únicamente los registros con extensión .xlsx para asegurarnos de mantener solo los adjuntos que sean ficheros Excel:



Añadimos el siguiente paso personalizado para obtener los datos de los ficheros Excel:

```
= Table.TransformColumns(#"Filas filtradas3", {"AttachmentContent", each Excel.Workbook(_, true)})
```

Expandimos la columna “AttachmentContent” seleccionando solo la columna “Data” y eliminamos el campo “Extension” pues no vamos a hacer uso de él:

	Subject	Name	Data
1	Test Data February 2023	Feb.xlsx	Table
2	Test Data January 2023	Jan.xlsx	Table

Por último, expandimos el campo “Data” y el resultado final con todos los registros combinados en la misma tabla es el siguiente:

Subject	Name	OrderDate	SalesOrderNumber	SalesAmount
Test Data February 2023	Feb.xlsx	01/02/2023	SO63316	4,99
Test Data February 2023	Feb.xlsx	02/02/2023	SO63317	4,99
Test Data February 2023	Feb.xlsx	03/02/2023	SO63313	4,99
Test Data February 2023	Feb.xlsx	04/02/2023	SO63355	539,99
Test Data February 2023	Feb.xlsx	05/02/2023	SO63352	2384,07
Test Data February 2023	Feb.xlsx	06/02/2023	SO63342	2294,99
Test Data February 2023	Feb.xlsx	07/02/2023	SO63335	769,49
Test Data February 2023	Feb.xlsx	08/02/2023	SO63319	21,98
Test Data February 2023	Feb.xlsx	09/02/2023	SO63306	28,99
Test Data February 2023	Feb.xlsx	10/02/2023	SO63316	120
Test Data February 2023	Feb.xlsx	11/02/2023	SO63317	54,99
Test Data February 2023	Feb.xlsx	12/02/2023	SO63319	54,99
Test Data February 2023	Feb.xlsx	13/02/2023	SO63313	8,99
Test Data February 2023	Feb.xlsx	14/02/2023	SO63352	8,99
Test Data February 2023	Feb.xlsx	15/02/2023	SO63355	8,99
Test Data February 2023	Feb.xlsx	16/02/2023	SO63335	34,99
Test Data February 2023	Feb.xlsx	17/02/2023	SO63306	4,99
Test Data February 2023	Feb.xlsx	18/02/2023	SO63342	4,99
Test Data February 2023	Feb.xlsx	19/02/2023	SO63352	4,99
Test Data February 2023	Feb.xlsx	20/02/2023	SO63355	4,99
Test Data February 2023	Feb.xlsx	21/02/2023	SO63306	8,99
Test Data February 2023	Feb.xlsx	22/02/2023	SO63319	7,95
Test Data February 2023	Feb.xlsx	23/02/2023	SO63342	9,99

2. Combinar ficheros Excel con encabezados diferentes

El propósito de este tip es mostrar cómo combinar ficheros de Excel cuyos encabezados son diferentes.

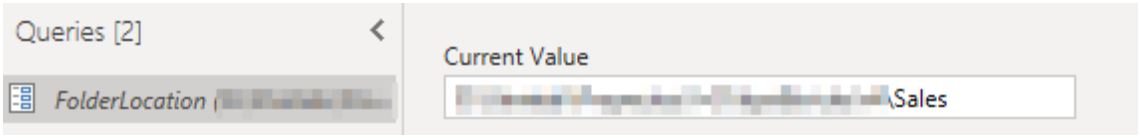


Desarrollo: En este caso, partimos de dos ficheros Excel que tienen la misma estructura, pero uno de los ficheros tiene los encabezados en inglés y el otro, en castellano.

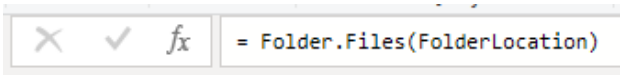
OrderDate	SalesOrderNumber	SalesAmount
01/01/2023	SO61294	4,99
02/01/2023	SO61302	2294,99
03/01/2023	SO61269	34,99
04/01/2023	SO61280	34,99
05/01/2023	SO61282	2,29
06/01/2023	SO61284	21,49
07/01/2023	SO61299	35
08/01/2023	SO61290	9,99
09/01/2023	SO61292	9,99
10/01/2023	SO61300	21,98
11/01/2023	SO61284	2,29
12/01/2023	SO61300	9,99
13/01/2023	SO61294	7,95
14/01/2023	SO61269	49,99
15/01/2023	SO61299	4,99
16/01/2023	SO61290	4,99
17/01/2023	SO61292	4,99
18/01/2023	SO61302	4,99
19/01/2023	SO61300	4,99
20/01/2023	SO61302	9,99
21/01/2023	SO61300	63,5
22/01/2023	SO61293	49,99
23/01/2023	SO61301	32,6
24/01/2023	SO61285	34,99
25/01/2023	SO61289	34,99
26/01/2023	SO61283	34,99
27/01/2023	SO61289	29,99
28/01/2023	SO61283	3,99
29/01/2023	SO61285	3,99
30/01/2023	SO61301	3,99
31/01/2023	SO61293	34,99

FechaPedido	NúmeroPedido	ImporteVentas
01/02/2023	SO63316	4,99
02/02/2023	SO63317	4,99
03/02/2023	SO63313	4,99
04/02/2023	SO63355	539,99
05/02/2023	SO63352	2384,07
06/02/2023	SO63342	2294,99
07/02/2023	SO63335	769,49
08/02/2023	SO63319	21,98
09/02/2023	SO63306	28,99
10/02/2023	SO63316	120
11/02/2023	SO63317	54,99
12/02/2023	SO63319	54,99
13/02/2023	SO63313	8,99
14/02/2023	SO63352	8,99
15/02/2023	SO63355	8,99
16/02/2023	SO63335	34,99
17/02/2023	SO63306	4,99
18/02/2023	SO63342	4,99
19/02/2023	SO63352	4,99
20/02/2023	SO63355	4,99
21/02/2023	SO63306	8,99
22/02/2023	SO63319	7,95
23/02/2023	SO63342	9,99
24/02/2023	SO63352	54,99
25/02/2023	SO63342	34,99
26/02/2023	SO63355	34,99
27/02/2023	SO63337	53,99
28/02/2023	SO63340	2,29

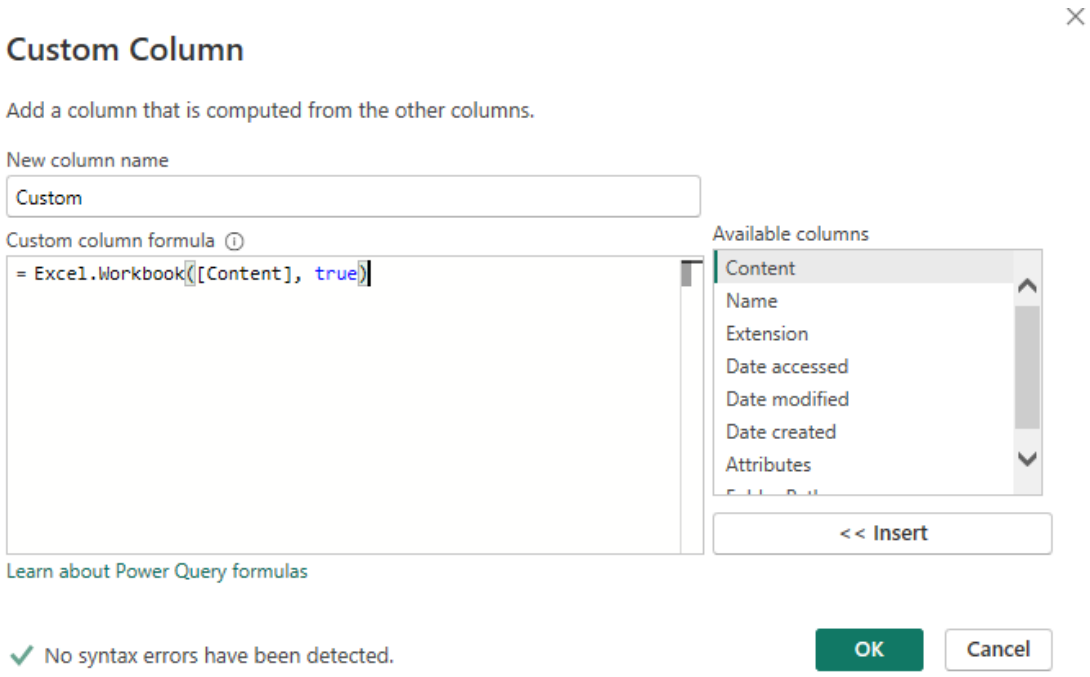
En primer lugar, creamos un parámetro con la ruta del fichero donde se encuentran los ficheros que queremos combinar.



Creamos una consulta en blanco utilizando la fórmula Folder.Files para obtener todos los ficheros localizados en esa carpeta:



Filtramos el campo “Extension” por el valor .xlsx para asegurarnos de que todos los archivos sean ficheros de Excel y añadimos una columna condicional para transformar los ficheros en formato tabular:



A continuación, expandimos todos los campos y hacemos Drill Down sobre la columna “Data” y luego añadimos la función Table.Combine para combinar las columnas de los dos ficheros:

```
= Table.Combine(#"Expanded Custom"[Data])
```

	FechaPedido	NúmeroPedido	ImporteVentas	OrderDate	SalesOrderNumber	SalesAmount
1	01/02/2023	SO63316	4,99	null	null	null
2	02/02/2023	SO63317	4,99	null	null	null
3	03/02/2023	SO63313	4,99	null	null	null
4	04/02/2023	SO63355	539,99	null	null	null
5	05/02/2023	SO63352	2384,07	null	null	null
6	06/02/2023	SO63342	2294,99	null	null	null
7	07/02/2023	SO63335	769,49	null	null	null
8	08/02/2023	SO63319	21,98	null	null	null
9	09/02/2023	SO63306	28,99	null	null	null
10	10/02/2023	SO63316	120	null	null	null
11	11/02/2023	SO63317	54,99	null	null	null
12	12/02/2023	SO63319	54,99	null	null	null
13	13/02/2023	SO63313	8,99	null	null	null
14	14/02/2023	SO63352	8,99	null	null	null
15	15/02/2023	SO63355	8,99	null	null	null
16	16/02/2023	SO63335	34,99	null	null	null
17	17/02/2023	SO63306	4,99	null	null	null
18	18/02/2023	SO63342	4,99	null	null	null
19	19/02/2023	SO63352	4,99	null	null	null
20	20/02/2023	SO63355	4,99	null	null	null
21	21/02/2023	SO63306	8,99	null	null	null
22	22/02/2023	SO63319	7,95	null	null	null
23	23/02/2023	SO63342	9,99	null	null	null
24	24/02/2023	SO63352	54,99	null	null	null
25	25/02/2023	SO63342	34,99	null	null	null
26	26/02/2023	SO63355	34,99	null	null	null
27	27/02/2023	SO63337	53,99	null	null	null
28	28/02/2023	SO63340	2,29	null	null	null
29	null	null	null	01/01/2023	SO61294	4,99
30	null	null	null	02/01/2023	SO61302	2294,99
31	null	null	null	03/01/2023	SO61269	34,99
32	null	null	null	04/01/2023	SO61280	34,99
33	null	null	null	05/01/2023	SO61282	2,29
34	null	null	null	06/01/2023	SO61284	21,49
35	null	null	null	07/01/2023	SO61299	35

El siguiente paso es añadir la función Table.ColumnNames para quedarnos solo con el nombre de los campos:

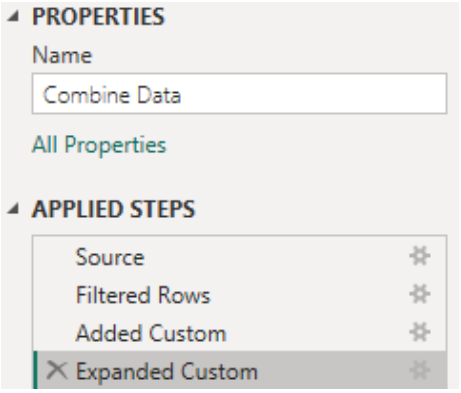
```
= Table.ColumnNames(Table.Combine(#"Expanded Custom"[Data]))
```

	List
1	FechaPedido
2	NúmeroPedido
3	ImporteVentas
4	OrderDate
5	SalesOrderNumber
6	SalesAmount

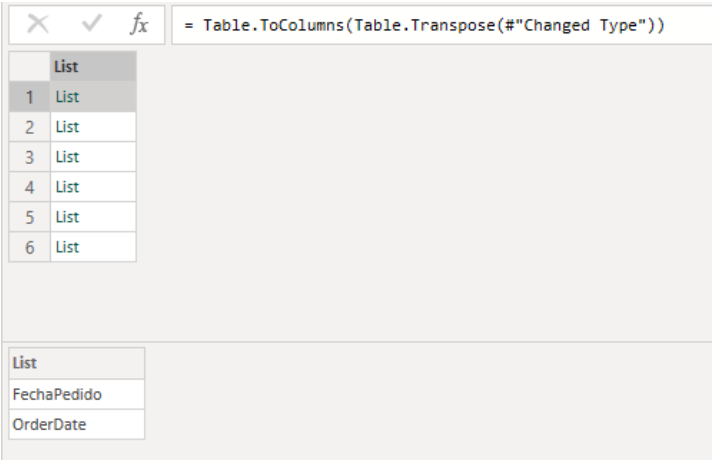
A continuación, creamos una tabla donde vamos a mapear los nombres en castellano con su correspondencia en inglés.

Original	Rename
FechaPedido	OrderDate
NúmeroPedido	SalesOrderNumber
ImporteVentas	SalesAmount
OrderDate	OrderDate
SalesOrderNumber	SalesOrderNumber
SalesAmount	SalesAmount

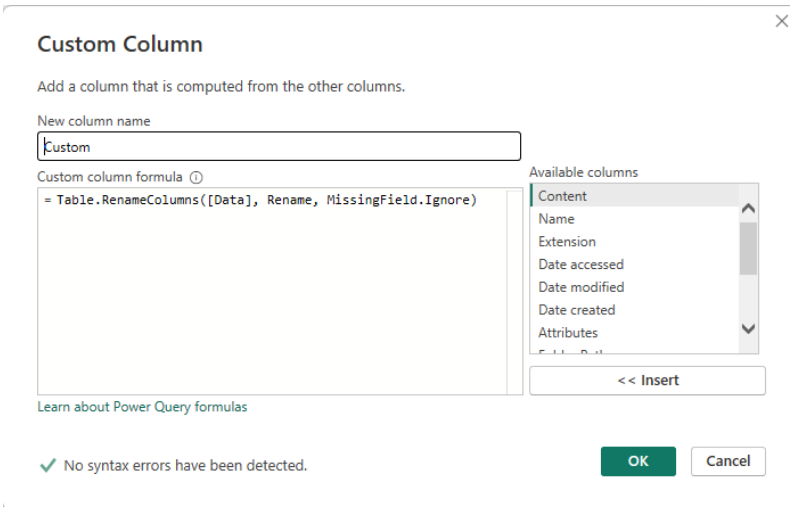
Duplicamos la tabla "Mapping", la renombramos a Combine Data y eliminamos todos los pasos excepto los 4 primeros:



Trasponemos la tabla "Rename" y utilizamos la función Table.ToColumns para convertir la tabla en una lista.

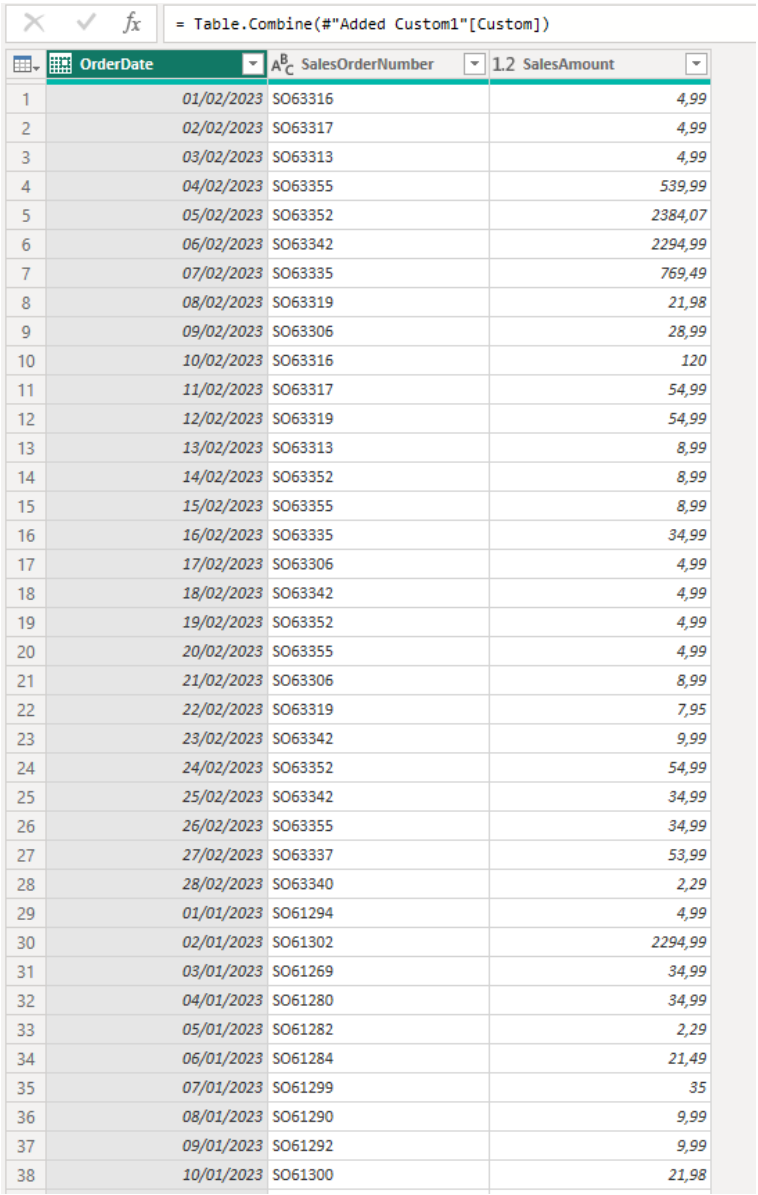


En la tabla “Combine Data” añadimos una nueva columna personalizada con la función Table.RenameColumns:



Seleccionamos la opción Drill Down sobre esta nueva columna para convertir este campo en una lista y utilizamos la función Table.Combine.

El resultado final con todos los registros de ambos ficheros combinados es el siguiente:



```
= Table.Combine(#"Added Custom1"[Custom])
```

	OrderDate	SalesOrderNumber	SalesAmount
1	01/02/2023	SO63316	4,99
2	02/02/2023	SO63317	4,99
3	03/02/2023	SO63313	4,99
4	04/02/2023	SO63355	539,99
5	05/02/2023	SO63352	2384,07
6	06/02/2023	SO63342	2294,99
7	07/02/2023	SO63335	769,49
8	08/02/2023	SO63319	21,98
9	09/02/2023	SO63306	28,99
10	10/02/2023	SO63316	120
11	11/02/2023	SO63317	54,99
12	12/02/2023	SO63319	54,99
13	13/02/2023	SO63313	8,99
14	14/02/2023	SO63352	8,99
15	15/02/2023	SO63355	8,99
16	16/02/2023	SO63335	34,99
17	17/02/2023	SO63306	4,99
18	18/02/2023	SO63342	4,99
19	19/02/2023	SO63352	4,99
20	20/02/2023	SO63355	4,99
21	21/02/2023	SO63306	8,99
22	22/02/2023	SO63319	7,95
23	23/02/2023	SO63342	9,99
24	24/02/2023	SO63352	54,99
25	25/02/2023	SO63342	34,99
26	26/02/2023	SO63355	34,99
27	27/02/2023	SO63337	53,99
28	28/02/2023	SO63340	2,29
29	01/01/2023	SO61294	4,99
30	02/01/2023	SO61302	2294,99
31	03/01/2023	SO61269	34,99
32	04/01/2023	SO61280	34,99
33	05/01/2023	SO61282	2,29
34	06/01/2023	SO61284	21,49
35	07/01/2023	SO61299	35
36	08/01/2023	SO61290	9,99
37	09/01/2023	SO61292	9,99
38	10/01/2023	SO61300	21,98

3. Alternativa a la función "Transponer"

El propósito de este tip es mostrar una alternativa a la función Transponer para los casos en los que el uso de dicha función no sea del todo eficiente.



Desarrollo: En el caso de que se necesite transponer los datos de una tabla con pocas filas y/o columnas, se podría utilizar la opción de "Transponer" de Power Query, pero en el caso de que la tabla tenga un gran volumen de datos, el uso de esta función hace que se ralentice la ejecución de la consulta.

Como alternativa, se pueden utilizar otras funciones con las que se pueden conseguir el mismo resultado.

En nuestro caso, partimos de la siguiente tabla:

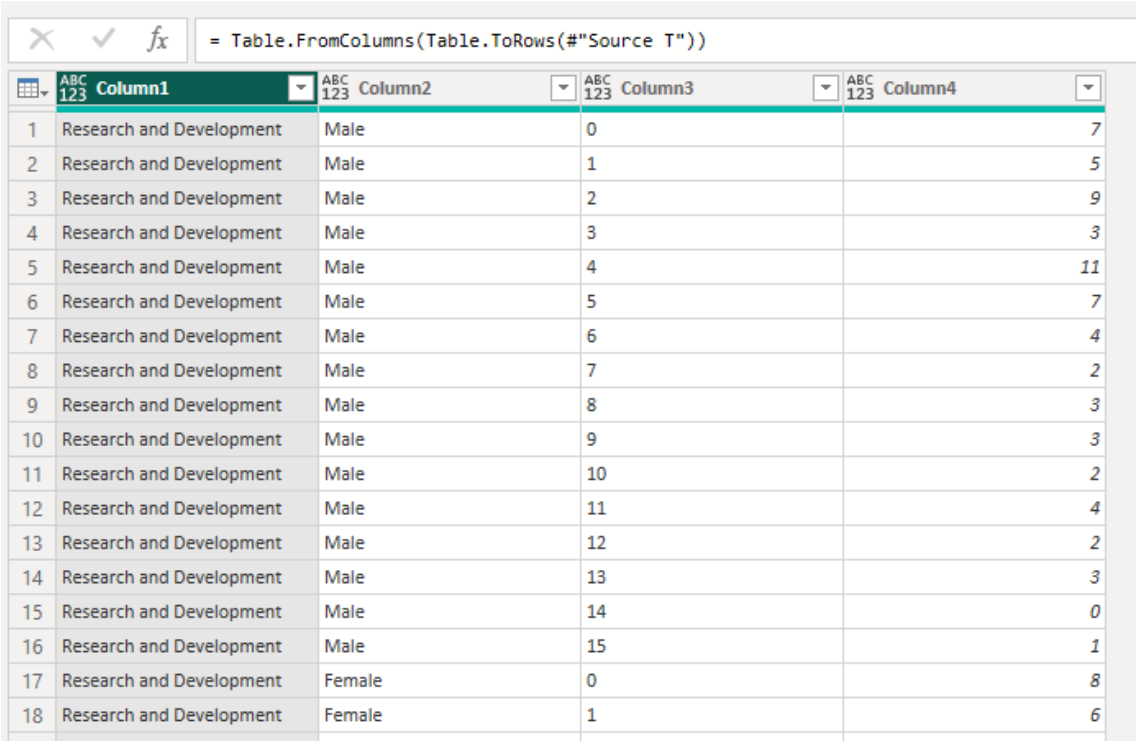
Column1	Column2	Column3	Column4	Column5	Column6
Research and Development	Research and Development	Research and Development	Research and Development	Research and Development	Research and D
Male	Male	Male	Male	Male	Male
0	1	2	3	4	5
7	5	9	3	11	

En primer lugar, vamos a convertir cada una de las filas en una lista y para ello, utilizamos la función Table.ToRows:

```
= Table.ToRows(#"Source T")
```

	List
1	List
2	List
3	List
4	List

A continuación, vamos a convertir cada lista en una columna con la función Table.FromColumns, obteniendo así la tabla transpuesta de la que original:



```
= Table.FromColumns(Table.ToRows(#"Source T"))
```

	Column1	Column2	Column3	Column4
1	Research and Development	Male	0	7
2	Research and Development	Male	1	5
3	Research and Development	Male	2	9
4	Research and Development	Male	3	3
5	Research and Development	Male	4	11
6	Research and Development	Male	5	7
7	Research and Development	Male	6	4
8	Research and Development	Male	7	2
9	Research and Development	Male	8	3
10	Research and Development	Male	9	3
11	Research and Development	Male	10	2
12	Research and Development	Male	11	4
13	Research and Development	Male	12	2
14	Research and Development	Male	13	3
15	Research and Development	Male	14	0
16	Research and Development	Male	15	1
17	Research and Development	Female	0	8
18	Research and Development	Female	1	6

4. Uso de registros como alternativa a crear columnas personalizadas

El propósito de este tip es ofrecer una alternativa a la creación de columnas personalizadas mediante el uso de registros.





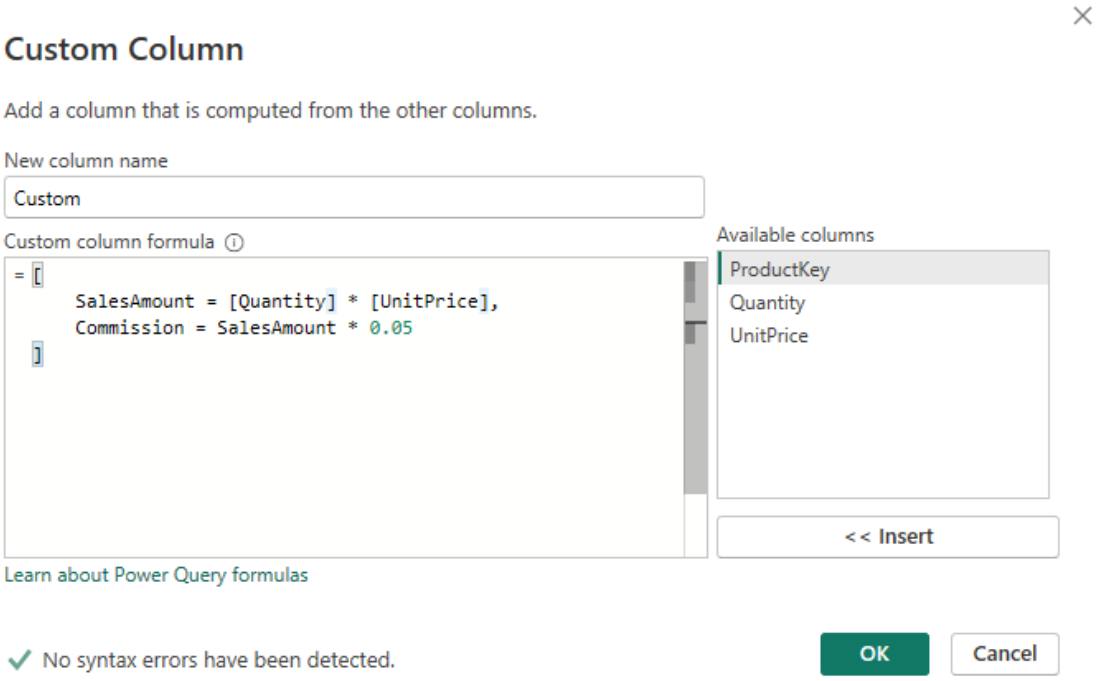
Desarrollo: En un escenario en el que se necesite crear más de una columna calculada, en vez de crearlas una a una, existe una alternativa que es crearlas a la vez mediante registros. En este tip veremos cómo hacerlo.

En este ejemplo, partiendo de las columnas “Quantity” y “UnitPrice” vamos a calcular el importe (SalesAmount) y una comisión sobre este.

Estos son los datos de partida:

ProductKey	Quantity	UnitPrice
477	5	4,99
530	10	4,99
604	6	539,99
584	6	539,99
606	10	539,99
359	8	2294,99
355	10	2319,99
355	3	2319,99
355	10	2319,99
388	2	1120,49
390	9	1120,49
390	2	1120,49
595	1	564,99
560	7	1214,85
485	1	21,98
485	4	21,98
485	9	21,98
478	2	9,99
537	9	35
529	4	3,99
541	7	28,99
539	4	24,99
482	10	8,99

En primer lugar, creamos una columna personalizada con los cálculos que necesitamos entre corchetes:



Con esto, obtenemos una columna con registros.

	ProductKey	Quantity	UnitPrice	Custom
1	477	5	4,99	Record
2	530	10	4,99	Record
3	604	6	539,99	Record
4	584	6	539,99	Record
5	606	10	539,99	Record
6	359	8	2294,99	Record
7	355	10	2319,99	Record
8	355	3	2319,99	Record
9	355	10	2319,99	Record
10	388	2	1120,49	Record
11	390	9	1120,49	Record
12	390	2	1120,49	Record
13	595	1	564,99	Record
14	560	7	1214,85	Record
15	485	1	21,98	Record
16	485	4	21,98	Record

El siguiente paso es expandir la columna y ya conseguimos el resultado deseado.

	ProductKey	Quantity	UnitPrice	SalesAmount	Commission
1	477	5	4,99	24,95	1,2475
2	530	10	4,99	49,9	2,495
3	604	6	539,99	3239,94	161,997
4	584	6	539,99	3239,94	161,997
5	606	10	539,99	5399,9	269,995
6	359	8	2294,99	18359,92	917,996
7	355	10	2319,99	23199,9	1159,995
8	355	3	2319,99	6959,97	347,9985
9	355	10	2319,99	23199,9	1159,995
10	388	2	1120,49	2240,98	112,049
11	390	9	1120,49	10084,41	504,2205
12	390	2	1120,49	2240,98	112,049
13	595	1	564,99	564,99	28,2495
14	560	7	1214,85	8503,95	425,1975
15	485	1	21,98	21,98	1,099
16	485	4	21,98	87,92	4,396
17	485	9	21,98	197,82	9,891
18	478	2	9,99	19,98	0,999
19	537	9	35	315	15,75
20	529	4	3,99	15,96	0,798

***Nota**: en este tip solo se han calculado dos columnas, pero se puede seguir el mismo planteamiento con el número de columnas que se necesite.

5. Dividir celdas con varias líneas de datos en diferentes registros

El propósito de este tip es mostrar cómo se pueden dividir celdas con varios valores separados en diferentes líneas en registros diferentes haciendo uso de la función de Power Query List.Zip.

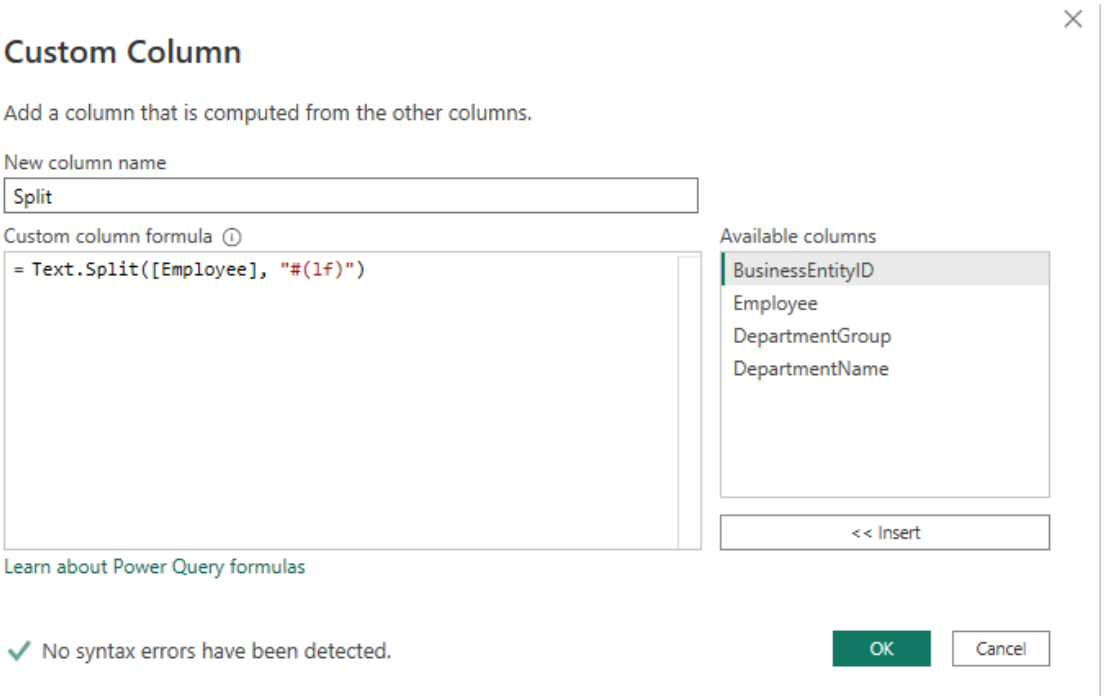


Desarrollo: Para este ejemplo, vamos a utilizar como fuente de datos un fichero Excel que contiene la siguiente tabla:

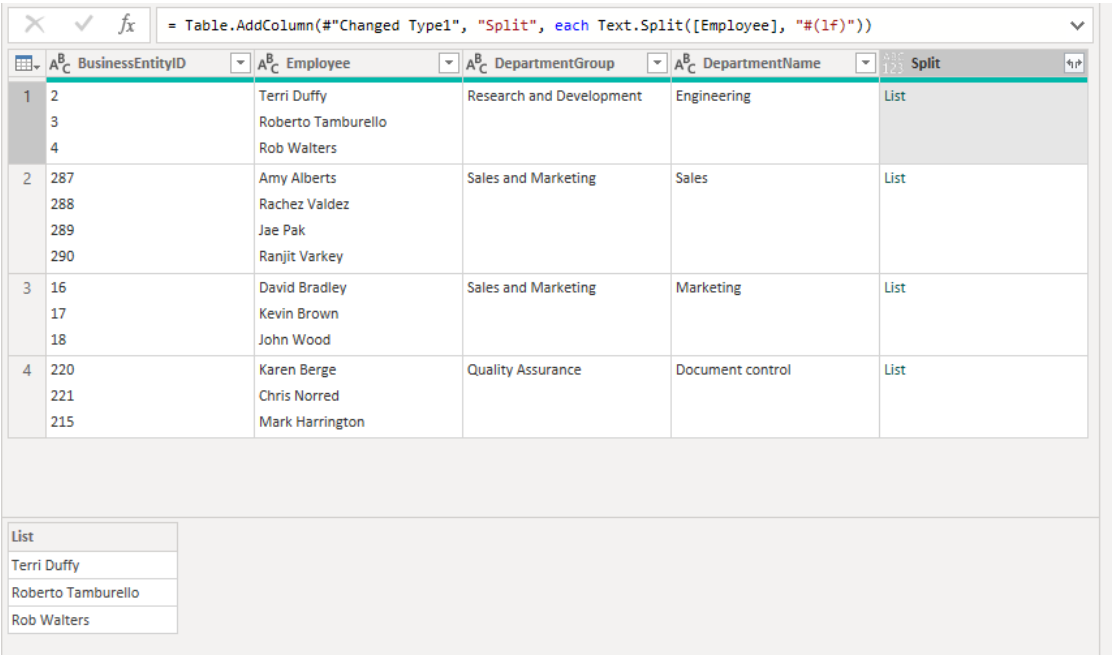
	BusinessEntityID	Employee	DepartmentGroup	DepartmentName
1	2	Terri Duffy	Research and Development	Engineering
	3	Roberto Tamburello		
	4	Rob Walters		
2	287	Amy Alberts	Sales and Marketing	Sales
	288	Rachez Valdez		
	289	Jae Pak		
	290	Ranjit Varkey		
3	16	David Bradley	Sales and Marketing	Marketing
	17	Kevin Brown		
	18	John Wood		
4	220	Karen Berge	Quality Assurance	Document control
	221	Chris Norred		
	215	Mark Harrington		

Como se puede ver en la imagen, las columnas “BusinessEntityID” y “Employee” tienen varios valores dentro de la misma celda. El objetivo es separar estos valores en registros diferentes.

El primer paso que se debe realizar es crear una nueva columna personalizada de forma que se cree una lista separando los valores de la celda por un carácter especial, en este caso, el final de línea.



De esta forma, tenemos una nueva columna cuyos valores son listas. Si seleccionamos una de estas celdas, podemos ver los valores de cada lista.



	BusinessEntityID	Employee	DepartmentGroup	DepartmentName	Split
1	2	Terri Duffy	Research and Development	Engineering	List
	3	Roberto Tamburello			
	4	Rob Walters			
2	287	Amy Alberts	Sales and Marketing	Sales	List
	288	Rachez Valdez			
	289	Jae Pak			
	290	Ranjit Varkey			
3	16	David Bradley	Sales and Marketing	Marketing	List
	17	Kevin Brown			
	18	John Wood			
4	220	Karen Berge	Quality Assurance	Document control	List
	221	Chris Norred			
	215	Mark Harrington			

List
Terri Duffy
Roberto Tamburello
Rob Walters

El siguiente paso consiste en crear un conjunto de valores basados en la posición que tengan en la lista. De esta forma, el primer valor de la lista 1 tiene que estar junto al primer valor de la lista 2, del primer valor de la lista 3, etc.

Para conseguir esto, se utiliza la función List.Zip, que combina los elementos que aparecen en diferentes listas, pero se encuentran en la misma posición, creando una nueva lista de valores.

Se agrega esta función en el paso de "Añadir Columna personalizada" de la siguiente forma:

```
= Table.AddColumn(#"Changed Type1", "Split", each List.Zip ({Text.Split([BusinessEntityID], "#(lf)"),
    Text.Split([Employee], "#(lf)")}))
```

Obteniendo una lista de listas:

	BusinessEntityID	Employee	DepartmentGroup	DepartmentName	Split
1	2 3 4	Terri Duffy Roberto Tamburello Rob Walters	Research and Development	Engineering	List
2	287 288 289 290	Amy Alberts Rachez Valdez Jae Pak Ranjit Varkey	Sales and Marketing	Sales	List
3	16 17 18	David Bradley Kevin Brown John Wood	Sales and Marketing	Marketing	List
4	220 221	Karen Berge Chris Norred	Quality Assurance	Document control	List

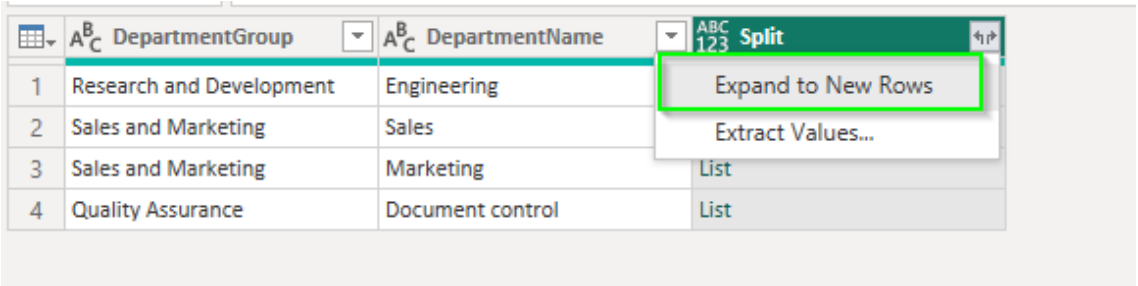
List
List
List
List

Se eliminan las columnas que contenían varios valores en la misma celda, puesto que se van a obtener de las listas creadas.

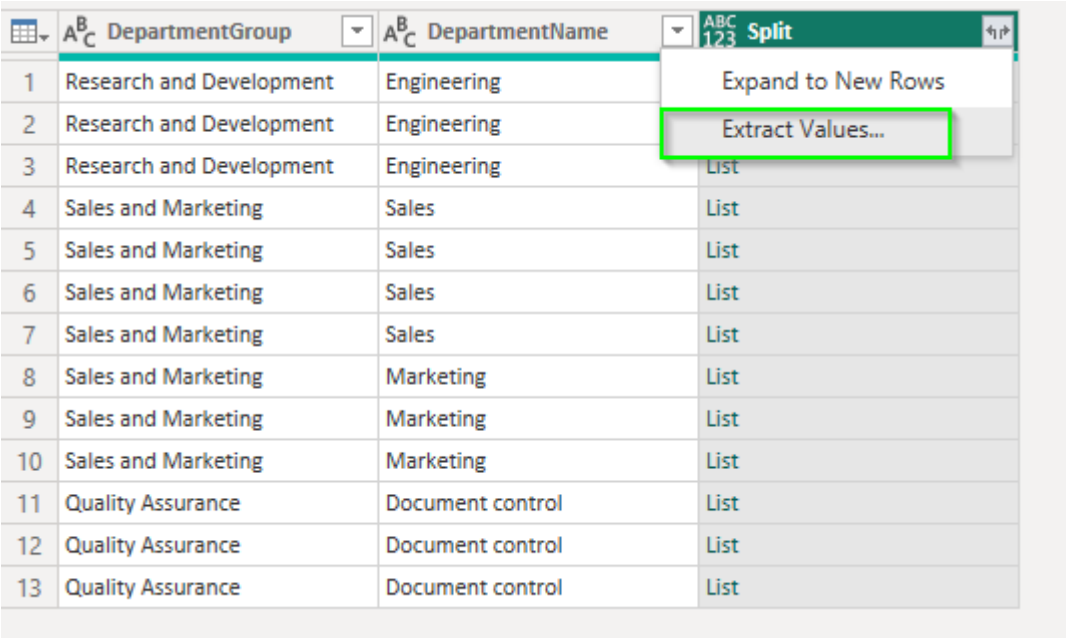
```
= Table.RemoveColumns(#"Added Custom",{"BusinessEntityID", "Employee"})
```

	DepartmentGroup	DepartmentName	Split
1	Research and Development	Engineering	List
2	Sales and Marketing	Sales	List
3	Sales and Marketing	Marketing	List
4	Quality Assurance	Document control	List

A continuación, se expande la lista creando nuevas filas:



El resultado que se obtiene es el siguiente y, a continuación, seleccionamos la opción de expandir valores. En este paso, se pedirá que se indique el delimitador entre los diferentes valores (en este caso, un punto y coma):



Obteniendo como resultado:

	DepartmentGroup	DepartmentName	Split
1	Research and Development	Engineering	2;Terri Duffy
2	Research and Development	Engineering	3;Roberto Tamburello
3	Research and Development	Engineering	4;Rob Walters
4	Sales and Marketing	Sales	287;Amy Alberts
5	Sales and Marketing	Sales	288;Rachez Valdez
6	Sales and Marketing	Sales	289;Jae Pak
7	Sales and Marketing	Sales	290;Ranjit Varkey
8	Sales and Marketing	Marketing	16;David Bradley
9	Sales and Marketing	Marketing	17;Kevin Brown
10	Sales and Marketing	Marketing	18;John Wood
11	Quality Assurance	Document control	220;Karen Berge
12	Quality Assurance	Document control	221;Chris Norred
13	Quality Assurance	Document control	215;Mark Harrington

Como último paso, se separa la columna Split por el delimitador punto y coma, se renombran y reordenan las columnas y el resultado final es el siguiente:

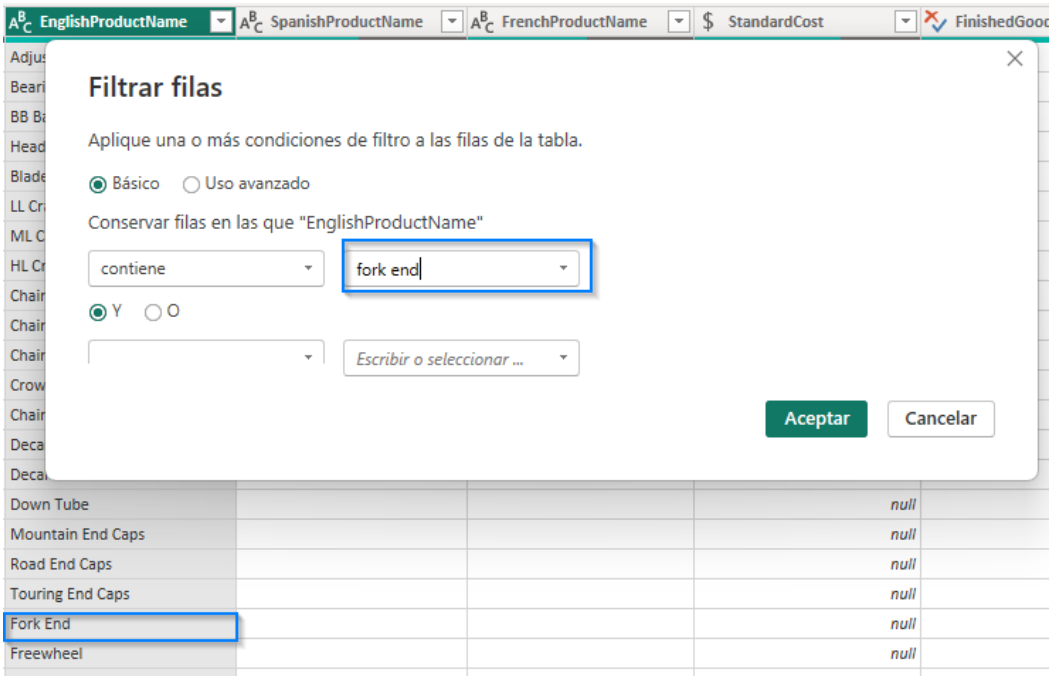
	BusinessEntityID	Employee	DepartmentGroup	DepartmentName
1	2	Terri Duffy	Research and Development	Engineering
2	3	Roberto Tamburello	Research and Development	Engineering
3	4	Rob Walters	Research and Development	Engineering
4	287	Amy Alberts	Sales and Marketing	Sales
5	288	Rachez Valdez	Sales and Marketing	Sales
6	289	Jae Pak	Sales and Marketing	Sales
7	290	Ranjit Varkey	Sales and Marketing	Sales
8	16	David Bradley	Sales and Marketing	Marketing
9	17	Kevin Brown	Sales and Marketing	Marketing
10	18	John Wood	Sales and Marketing	Marketing
11	220	Karen Berge	Quality Assurance	Document control
12	221	Chris Norred	Quality Assurance	Document control
13	215	Mark Harrington	Quality Assurance	Document control

6. Ignorar case sensitivity

El propósito de este tip es mostrar cómo deshabilitar el case sensitivity de Power Query.



Desarrollo: Como Power Query es case sensitive por defecto, si intentamos filtrar por un valor que no tenga las mayúsculas y minúsculas exactamente igual que el valor que aparece en la tabla, Power Query va a mostrar 0 resultados.



Para hacer que no sea case sensitive, tenemos que añadir lo siguiente en el paso de filas filtradas:



7. Expandir las columnas sin especificar el nombre de estas

El propósito de este tip es mostrar cómo se pueden expandir de forma automática las columnas de una tabla sin especificar el nombre de las mismas.



Desarrollo: Partiendo de una tabla con un campo agrupado, si seleccionamos el icono  se van a expandir los campos, pero en el propio paso aparecen los nombres de las columnas que se han expandido y, por tanto, cualquier modificación en el nombre o el número de las columnas, va a hacer que la query se rompa mostrando un error.

	EnglishCountryRegionName	All
1	Australia	Table
2	Canada	Table
3	Germany	Table
4	France	Table
5	United Kingdom	Table
6	United States	Table

```
= Table.ExpandTableColumn(#"Filas agrupadas", "All", {"GeographyKey", "City", "StateProvinceCode",
    "StateProvinceName", "CountryRegionCode", "EnglishCountryRegionName", "SpanishCountryRegionName",
    "FrenchCountryRegionName", "PostalCode", "SalesTerritoryKey", "DimCustomer", "DimReseller", "DimSalesTerritory"},
    {"GeographyKey", "City", "StateProvinceCode", "StateProvinceName", "CountryRegionCode",
    "EnglishCountryRegionName.1", "SpanishCountryRegionName", "FrenchCountryRegionName", "PostalCode",
```

El objetivo es que se expandan las columnas de forma dinámica sin que se especifique en ningún paso el nombre de las columnas expandidas.

Para ello, primero vamos a crear un nuevo paso justo antes del paso de Expandir las columnas:

```
= Table.ColumnNames(Table.Combine(#"Filas agrupadas" [All]))
```

Con esto conseguimos en forma de lista el nombre de todos los campos que aparecen en la tabla.

	Lista
1	GeographyKey
2	City
3	StateProvinceCode
4	StateProvinceName
5	CountryRegionCode
6	EnglishCountryRegionName
7	SpanishCountryRegionName
8	FrenchCountryRegionName
9	PostalCode
10	SalesTerritoryKey
11	DimCustomer
12	DimReseller
13	DimSalesTerritory

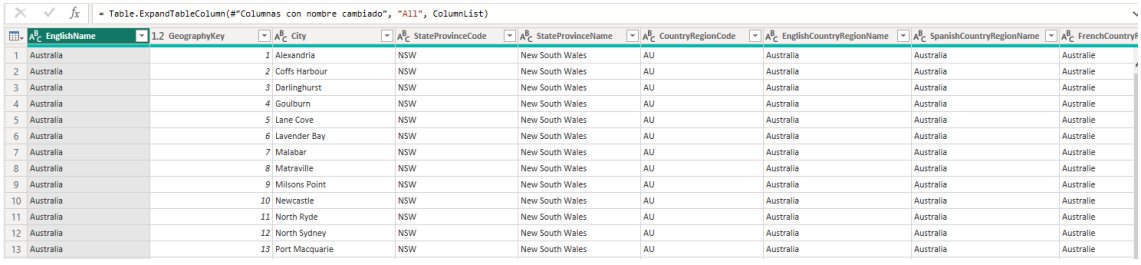
Antes de modificar el paso de Expandir Columnas, vamos a cambiar el nombre del campo EnglishCountryRegionName a EnglishName, ya que si lo dejamos igual, al expandir las columnas, nos saldrá un error porque habría dos campos con el mismo nombre.

A continuación, volvemos al paso de Expandir Columnas y lo modificamos de la siguiente manera:

```
= Table.ExpandTableColumn(#"Filas agrupadas", "All", ColumnList)
```

ColumnList es el nombre que se le ha asignado al paso añadido.

Con esto, ya tenemos todos los campos expandidos sin necesidad de codificar los nombres.



	EnglishName	GeographyKey	City	StateProvinceCode	StateProvinceName	CountryRegionCode	EnglishCountryRegionName	SpanishCountryRegionName	FrenchCountryR
1	Australia	1	Alexandria	NSW	New South Wales	AU	Australia	Australia	Australie
2	Australia	2	Coffs Harbour	NSW	New South Wales	AU	Australia	Australia	Australie
3	Australia	3	Darlinghurst	NSW	New South Wales	AU	Australia	Australia	Australie
4	Australia	4	Goulburn	NSW	New South Wales	AU	Australia	Australia	Australie
5	Australia	5	Lane Cove	NSW	New South Wales	AU	Australia	Australia	Australie
6	Australia	6	Lavender Bay	NSW	New South Wales	AU	Australia	Australia	Australie
7	Australia	7	Malabar	NSW	New South Wales	AU	Australia	Australia	Australie
8	Australia	8	Matraville	NSW	New South Wales	AU	Australia	Australia	Australie
9	Australia	9	Milsons Point	NSW	New South Wales	AU	Australia	Australia	Australie
10	Australia	10	Newcastle	NSW	New South Wales	AU	Australia	Australia	Australie
11	Australia	11	North Ryde	NSW	New South Wales	AU	Australia	Australia	Australie
12	Australia	12	North Sydney	NSW	New South Wales	AU	Australia	Australia	Australie
13	Australia	13	Port Macquarie	NSW	New South Wales	AU	Australia	Australia	Australie

8. Desapilar filas en columnas

El propósito de este tip es mostrar cómo convertir una columna de datos en las correspondientes columnas.





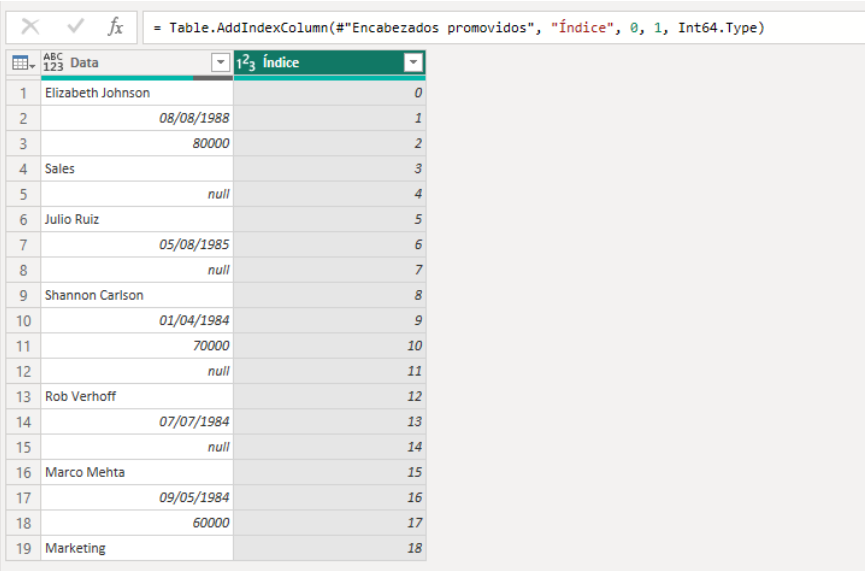
Desarrollo: Este ejemplo se desarrollará partiendo de un fichero Excel con la siguiente estructura:

A
Elizabeth Johnson
08/08/1988
80000
Sales
Julio Ruiz
05/08/1985
Shannon Carlson
01/04/1984
70000
Rob Verhoff
07/07/1984
Marco Mehta
09/05/1984
60000
Marketing

Las diferentes filas están separadas por una celda en blanco en la columna inicial. Como podemos observar, hay filas que tienen datos correspondientes a 4 columnas mientras que otras solo tienen datos de 2 o 3 columnas.

Una vez cargado el fichero en Power BI, el primer paso a realizar es añadir una columna de índice.

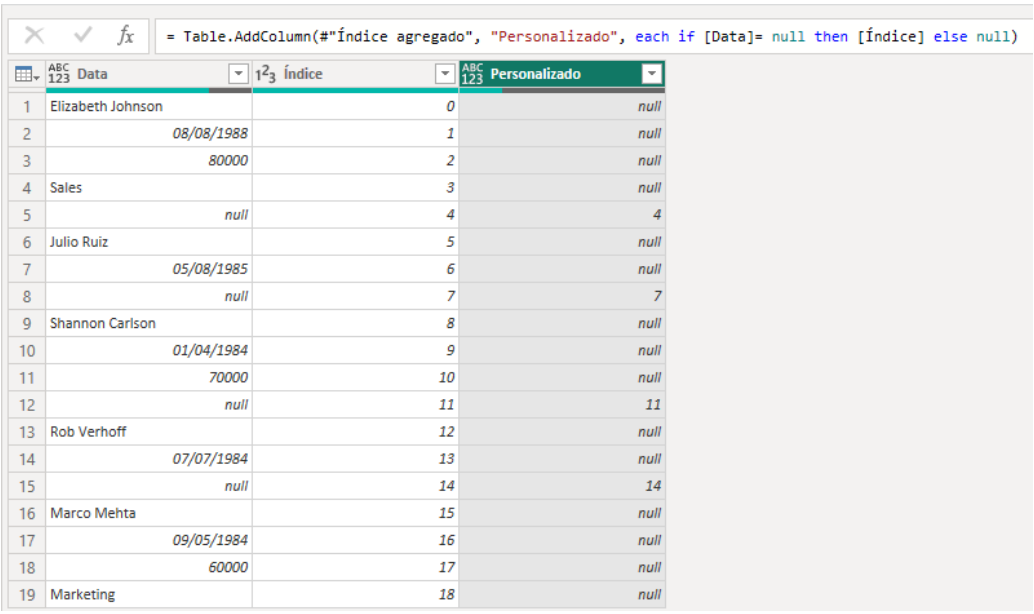
```
= Table.AddIndexColumn(#"Encabezados promovidos", "Índice", 0, 1, Int64.Type)
```



	Data	Índice
1	Elizabeth Johnson	0
2	08/08/1988	1
3	80000	2
4	Sales	3
5	null	4
6	Julio Ruiz	5
7	05/08/1985	6
8	null	7
9	Shannon Carlson	8
10	01/04/1984	9
11	70000	10
12	null	11
13	Rob Verhoff	12
14	07/07/1984	13
15	null	14
16	Marco Mehta	15
17	09/05/1984	16
18	60000	17
19	Marketing	18

A continuación, añadimos la siguiente columna personalizada:

```
= Table.AddColumn(#"Índice agregado", "Personalizado", each if [Data]= null then [Índice] else null)
```



	Data	Índice	Personalizado
1	Elizabeth Johnson	0	null
2	08/08/1988	1	null
3	80000	2	null
4	Sales	3	null
5	null	4	4
6	Julio Ruiz	5	null
7	05/08/1985	6	null
8	null	7	7
9	Shannon Carlson	8	null
10	01/04/1984	9	null
11	70000	10	null
12	null	11	11
13	Rob Verhoff	12	null
14	07/07/1984	13	null
15	null	14	14
16	Marco Mehta	15	null
17	09/05/1984	16	null
18	60000	17	null
19	Marketing	18	null

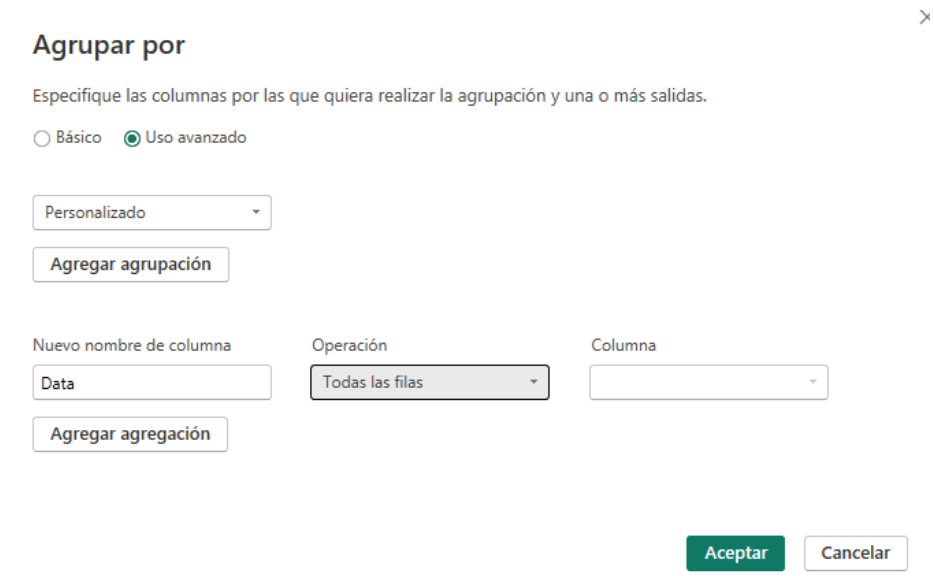
Luego, aplicamos la transformación Table.FillUp para sustituir los valores nulos de la columna personalizada por los valores correspondientes.

	Data	Índice	Personalizado
1	Elizabeth Johnson	0	4
2	08/08/1988	1	4
3	80000	2	4
4	Sales	3	4
5	null	4	4
6	Julio Ruiz	5	7
7	05/08/1985	6	7
8	null	7	7
9	Shannon Carlson	8	11
10	01/04/1984	9	11
11	70000	10	11
12	null	11	11
13	Rob Verhoff	12	14
14	07/07/1984	13	14
15	null	14	14
16	Marco Mehta	15	19
17	09/05/1984	16	19
18	60000	17	19
19	Marketing	18	19
20	null	19	19

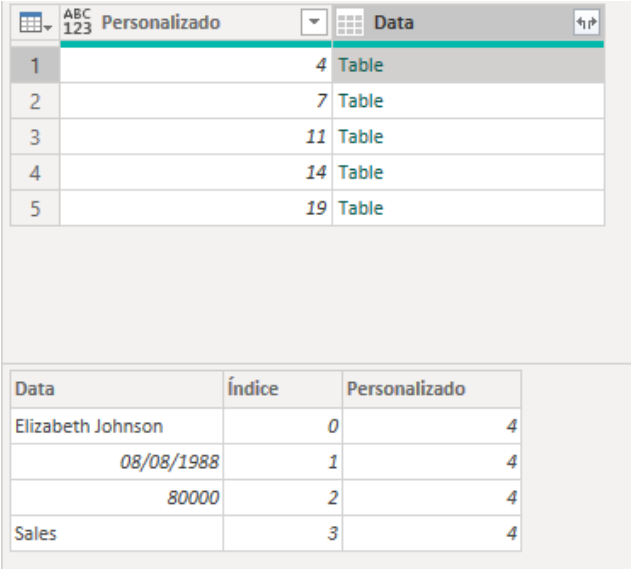
Posteriormente, filtramos el campo Data por los registros que tengan valores distintos de null.

	Data	Índice	Personalizado
1	Elizabeth Johnson	0	4
2	08/08/1988	1	4
3	80000	2	4
4	Sales	3	4
5	Julio Ruiz	5	7
6	05/08/1985	6	7
7	Shannon Carlson	8	11
8	01/04/1984	9	11
9	70000	10	11
10	Rob Verhoff	12	14
11	07/07/1984	13	14
12	Marco Mehta	15	19
13	09/05/1984	16	19
14	60000	17	19
15	Marketing	18	19

A continuación, agrupamos los valores por el campo “Personalizado”.



El resultado es:



	Personalizado	Data
1	4	Table
2	7	Table
3	11	Table
4	14	Table
5	19	Table

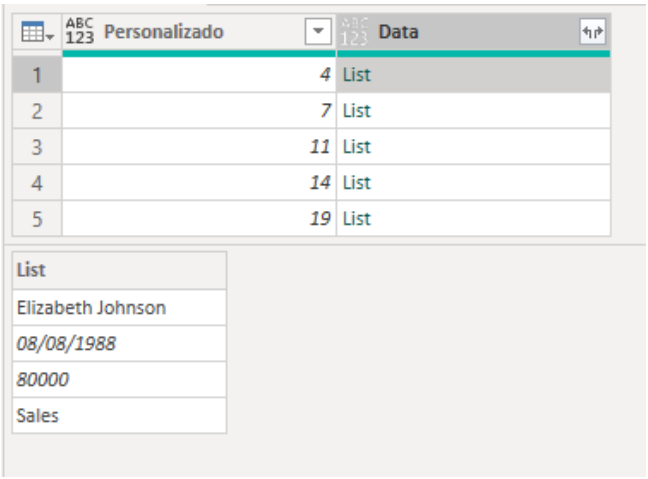
Data	Índice	Personalizado
Elizabeth Johnson	0	4
08/08/1988	1	4
80000	2	4
Sales	3	4

Luego, modificamos el paso para convertir las tablas en listas de la siguiente forma:

```
= Table.Group(#"Filas filtradas", {"Personalizado"}, {{"Data", each _, type table [Data=any, Índice=number,
    Personalizado=number]}})
```



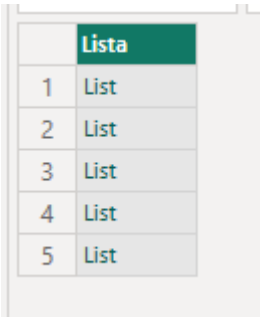
```
= Table.Group(#"Filas filtradas", {"Personalizado"}, {{"Data", each _[Data]}})
```



	Personalizado	Data
1	4	List
2	7	List
3	11	List
4	14	List
5	19	List

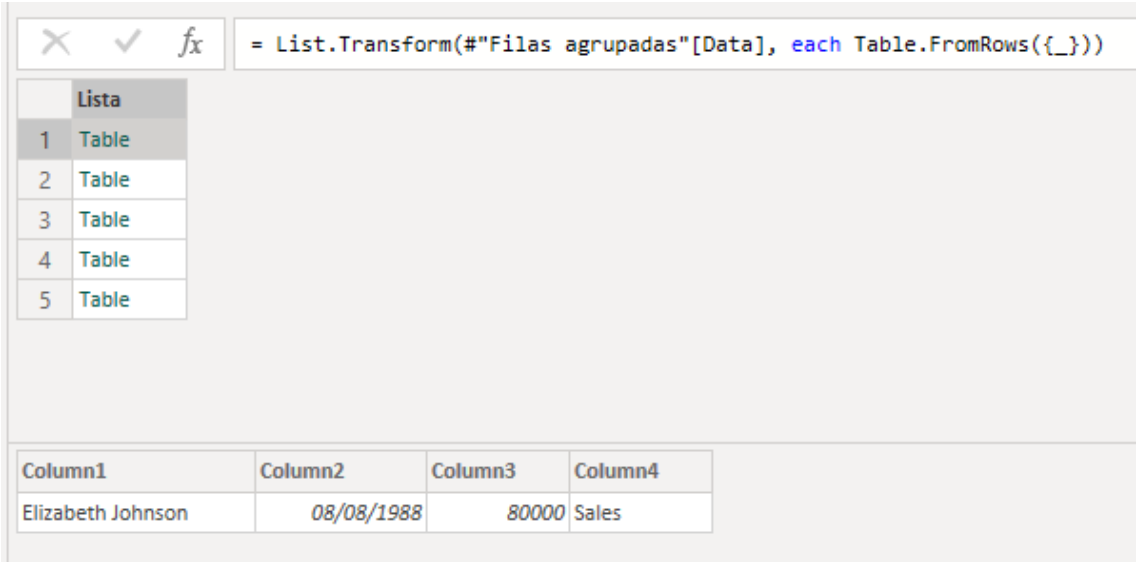
List
Elizabeth Johnson
08/08/1988
80000
Sales

Ahora, hacemos “Drill down” sobre la columna “Data” (haciendo click derecho sobre la columna, el nombre en castellano es “Rastrear desagrupando datos”), obteniendo una lista de las listas que teníamos:



	Lista
1	List
2	List
3	List
4	List
5	List

Editamos este último paso para convertir cada una de las listas en una tabla con las columnas correspondientes:

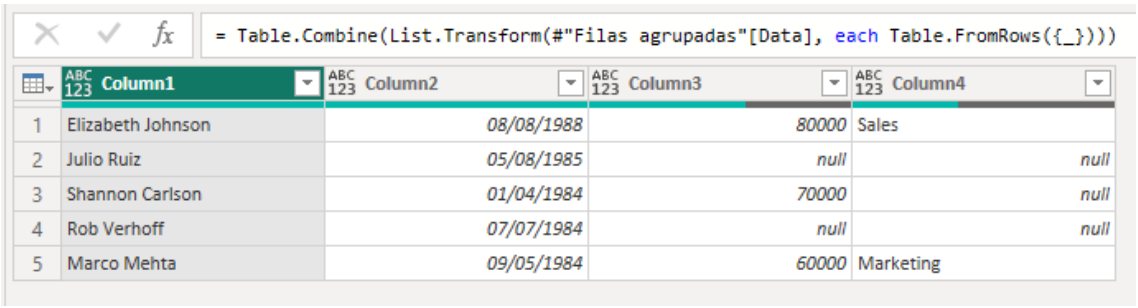


```
= List.Transform(#"Filas agrupadas"[Data], each Table.FromRows({_}))
```

	Lista
1	Table
2	Table
3	Table
4	Table
5	Table

Column1	Column2	Column3	Column4
Elizabeth Johnson	*08/08/1988*	*80000*	Sales

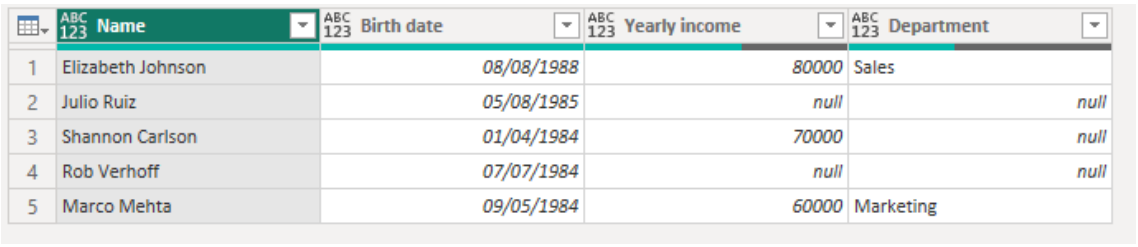
Para combinar todas las tablas, volvemos a editar el paso anterior añadiendo la función Table.Combine:



```
= Table.Combine(List.Transform(#"Filas agrupadas"[Data], each Table.FromRows({_})))
```

	Column1	Column2	Column3	Column4
1	Elizabeth Johnson	*08/08/1988*	*80000*	Sales
2	Julio Ruiz	*05/08/1985*	*null*	*null*
3	Shannon Carlson	*01/04/1984*	*70000*	*null*
4	Rob Verhoff	*07/07/1984*	*null*	*null*
5	Marco Mehta	*09/05/1984*	*60000*	Marketing

Por último, renombramos los campos y el resultado final es:



	Name	Birth date	Yearly income	Department
1	Elizabeth Johnson	*08/08/1988*	*80000*	Sales
2	Julio Ruiz	*05/08/1985*	*null*	*null*
3	Shannon Carlson	*01/04/1984*	*70000*	*null*
4	Rob Verhoff	*07/07/1984*	*null*	*null*
5	Marco Mehta	*09/05/1984*	*60000*	Marketing

2. MODELADO

9. Convertir matriz en datos tabulares

El propósito de este tip es mostrar cómo se puede convertir una matriz de Excel en formato tabular para poder analizar los datos desde Power BI.



Desarrollo: Este ejemplo se desarrollará partiendo de un fichero Excel con la siguiente estructura:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Number of employees by year worked in the company, gender, and department																		
2														Years worked in the company					
3			Total	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	Research and																		
5		Total	128	15	11	16	4	20	15	7	6	5	4	4	7	3	7	2	2
6		Male	66	7	5	9	3	11	7	4	2	3	3	2	4	2	3	0	1
7		Female	62	8	6	7	1	9	8	3	4	2	1	2	3	1	4	2	1
8	Sales and Marketing																		
9		Total	128	8	11	13	12	10	6	10	20	14	15	9	0	0	0	0	0
10		Male	61	1	5	3	5	4	2	7	12	5	8	9	0	0	0	0	0
11		Female	67	7	6	10	7	6	4	3	8	9	7	0	0	0	0	0	0
12	Human resources																		
13		Total	116	3	15	2	18	7	11	17	7	7	9	10	4	5	1	0	0
14		Male	58	1	6	0	8	4	5	10	2	6	7	4	2	2	1	0	0
15		Female	58	2	9	2	10	3	6	7	5	1	2	6	2	3	0	0	0
16	Quality Assurance																		
17		Total	137	11	0	7	8	5	23	20	25	15	15	5	0	3	0	0	0
18		Male	76	8	0	0	8	5	9	14	13	6	8	5	0	0	0	0	0
19		Female	61	3	0	7	0	0	14	6	12	9	7	0	0	3	0	0	0
20																			

Como no es posible analizar los datos con esta disposición, el objetivo es convertir la matriz anterior en la siguiente tabla:

Department	Gender	Year Group	Value
Research and Development	Male	0-5	42
Research and Development	Male	6-10	14
Research and Development	Male	10-15	10
Research and Development	Female	0-5	39
Research and Development	Female	6-10	12
Research and Development	Female	10-15	11
Sales and Marketing	Male	0-5	20
Sales and Marketing	Male	6-10	41
Sales and Marketing	Male	10-15	0
Sales and Marketing	Female	0-5	40
Sales and Marketing	Female	6-10	27
Sales and Marketing	Female	10-15	0
Human resources	Male	0-5	24
Human resources	Male	6-10	29
Human resources	Male	10-15	5
Human resources	Female	0-5	32
Human resources	Female	6-10	21
Human resources	Female	10-15	5
Quality Assurance	Male	0-5	30
Quality Assurance	Male	6-10	46
Quality Assurance	Male	10-15	0
Quality Assurance	Female	0-5	24
Quality Assurance	Female	6-10	34
Quality Assurance	Female	10-15	3

Una vez cargado el fichero Excel en Power BI, desde Power Query veremos lo siguiente:

	Number of employees by...	Column2	Column3	Column4	Column5	Column6	Column7
1	null	null	Years worked in the company	null	null	null	r
2	null	null	Total	0	1	2	
3	Research and Development	null	null	null	null	null	r
4	null	Total	128	15	11	16	
5	null	Male	66	7	5	9	
6	null	Female	62	8	6	7	
7	Sales and Marketing	null	null	null	null	null	r
8	null	Total	128	8	11	13	
9	null	Male	61	1	5	3	
10	null	Female	67	7	6	10	
11	Human resources	null	null	null	null	null	r
12	null	Total	116	3	15	2	
13	null	Male	58	1	6	0	
14	null	Female	58	2	9	2	
15	Quality Assurance	null	null	null	null	null	r
16	null	Total	137	11	0	7	
17	null	Male	76	8	0	0	
18	null	Female	61	3	0	7	

En primer lugar, vamos a eliminar la primera fila, ya que no tiene datos de análisis. A continuación, haremos uso de la función Table.FillDown sobre el campo “Number of employees by..” ya que cada categoría se repite 3 veces (Total, Male, Female). Para ello, añadiremos un paso personalizado de la siguiente forma, siendo el resultado de esta transformación el que se muestra en la captura siguiente:

```
= Table.FillDown(#"Filas superiores quitadas", {"Number of employees by year worked in the company, gender, and department"})
```

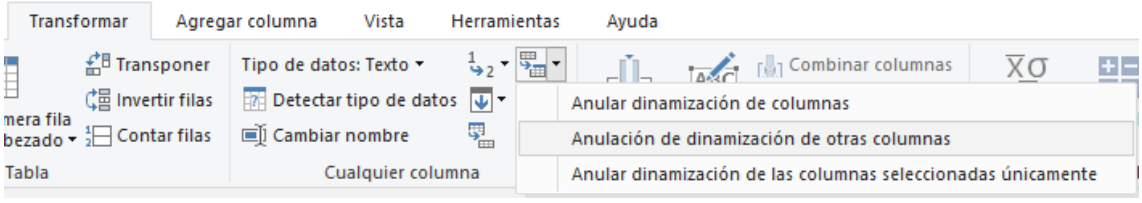
Number of employees by...	Column2	Column3	Colu
null	null	Total	
Research and Development	null	null	
Research and Development	Total	128	
Research and Development	Male	66	
Research and Development	Female	62	
Sales and Marketing	null	null	
Sales and Marketing	Total	128	
Sales and Marketing	Male	61	
Sales and Marketing	Female	67	
Human resources	null	null	
Human resources	Total	116	
Human resources	Male	58	
Human resources	Female	58	
Quality Assurance	null	null	
Quality Assurance	Total	137	
Quality Assurance	Male	76	
Quality Assurance	Female	61	

A continuación, aplicamos la transformación de “Usar primera fila como encabezado” y renombramos la primera columna como “Department” y la segunda, como “Gender”.

	Department	Gender	Total	0	1	2	3	
1	Research and Development	null	null	null	null	null	null	
2	Research and Development	Total	128	15	11	16	4	
3	Research and Development	Male	66	7	5	9	3	
4	Research and Development	Female	62	8	6	7	1	
5	Sales and Marketing	null	null	null	null	null	null	
6	Sales and Marketing	Total	128	8	11	13	12	
7	Sales and Marketing	Male	61	1	5	3	5	
8	Sales and Marketing	Female	67	7	6	10	7	
9	Human resources	null	null	null	null	null	null	
10	Human resources	Total	116	3	15	2	18	
11	Human resources	Male	58	1	6	0	8	
12	Human resources	Female	58	2	9	2	10	
13	Quality Assurance	null	null	null	null	null	null	
14	Quality Assurance	Total	137	11	0	7	8	
15	Quality Assurance	Male	76	8	0	0	8	
16	Quality Assurance	Female	61	3	0	7	0	

Eliminamos la columna llamada “Total” y filtramos las filas que tienen el valor “Total” y null en el campo de “Gender”.

Seleccionamos las columnas Department y Gender y aplicamos la transformación “Anulación de dinamización de otras columnas”, obteniendo el siguiente resultado:



Department	Gender	Atributo	Valor
Research and Development	Male	0	7
Research and Development	Male	1	5
Research and Development	Male	2	9
Research and Development	Male	3	3
Research and Development	Male	4	11
Research and Development	Male	5	7
Research and Development	Male	6	4
Research and Development	Male	7	2
Research and Development	Male	8	3
Research and Development	Male	9	3
Research and Development	Male	10	2
Research and Development	Male	11	4
Research and Development	Male	12	2
Research and Development	Male	13	3
Research and Development	Male	14	0
Research and Development	Male	15	1
Research and Development	Female	0	8
Research and Development	Female	1	6
Research and Development	Female	2	7
Research and Development	Female	3	1
Research and Development	Female	4	9
Research and Development	Female	5	8
Research and Development	Female	6	3

Así, tenemos las columnas despivotadas de forma que el campo Atributo hace referencia a los años trabajados en la empresa y en el campo Valor, el número de trabajadores.

Renombrando estos dos campos, el resultado final es el siguiente:

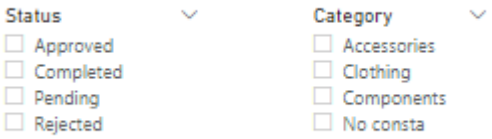
	Department	Gender	Years of experience	Count
28	Research and Development	Female	11	3
29	Research and Development	Female	12	1
30	Research and Development	Female	13	4
31	Research and Development	Female	14	2
32	Research and Development	Female	15	1
33	Sales and Marketing	Male	0	1
34	Sales and Marketing	Male	1	5
35	Sales and Marketing	Male	2	3
36	Sales and Marketing	Male	3	5
37	Sales and Marketing	Male	4	4
38	Sales and Marketing	Male	5	2
39	Sales and Marketing	Male	6	7
40	Sales and Marketing	Male	7	12
41	Sales and Marketing	Male	8	5
42	Sales and Marketing	Male	9	8
43	Sales and Marketing	Male	10	9
44	Sales and Marketing	Male	11	0
45	Sales and Marketing	Male	12	0
46	Sales and Marketing	Male	13	0
47	Sales and Marketing	Male	14	0
48	Sales and Marketing	Male	15	0
49	Sales and Marketing	Female	0	7
50	Sales and Marketing	Female	1	6

10. Filtrar valores entre segmentaciones

El objetivo de este tip es mostrar cómo poder filtrar valores entre diferentes segmentaciones de forma que cuando se seleccione uno o varios valores de una, solo se muestren los valores correspondientes de la otra segmentación.



Desarrollo: En nuestro ejemplo, la dimensión Status y la dimensión Product no están relacionadas entre sí. El objetivo es que se puedan filtrar entre sí de forma que, al seleccionar un valor concreto de Status, solo se muestren las categorías de los productos que tienen ese status y viceversa.

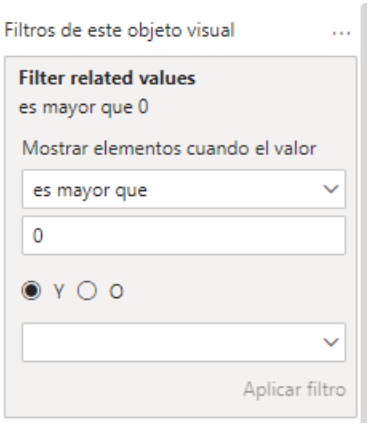


En el fondo, lo que vamos a crear es una relación entre ambas dimensiones para que puedan interactuar entre ellas. Para ello, vamos a crear una nueva medida llamada Filter related values de la siguiente forma:

```
1  Filter related values = COUNTROWS(Fact_purchases)
```

En este caso, como el modelo solo contiene una tabla de hechos, solo se contabilizan las filas de esta tabla. En el caso de modelos más complejos con varias tablas de hechos, esta medida debe sumar el número de filas de todas las tablas de hechos involucradas.

A continuación, añadimos a cada segmentación el siguiente filtro para conseguir el comportamiento deseado:



Finalmente, podemos observar cómo al seleccionar un valor de Status, se muestran solo las categorías que tienen ese status y viceversa:



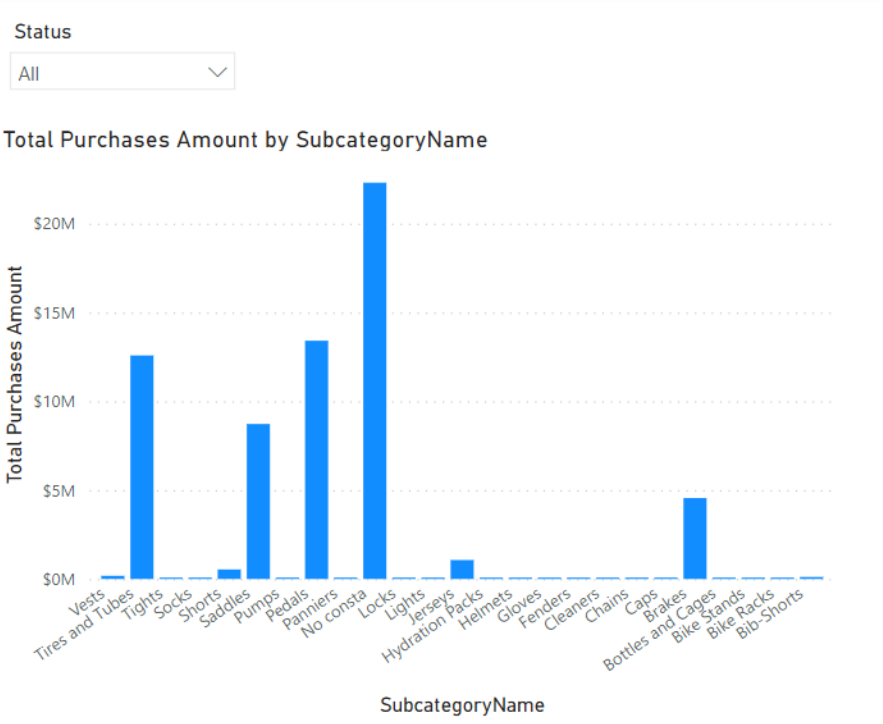
11. Rango estático de los ejes

El objetivo de este tip es mostrar cómo definir el rango de los ejes de un gráfico independientemente de la selección que haya en las segmentaciones, manteniendo los ejes con un rango fijo.

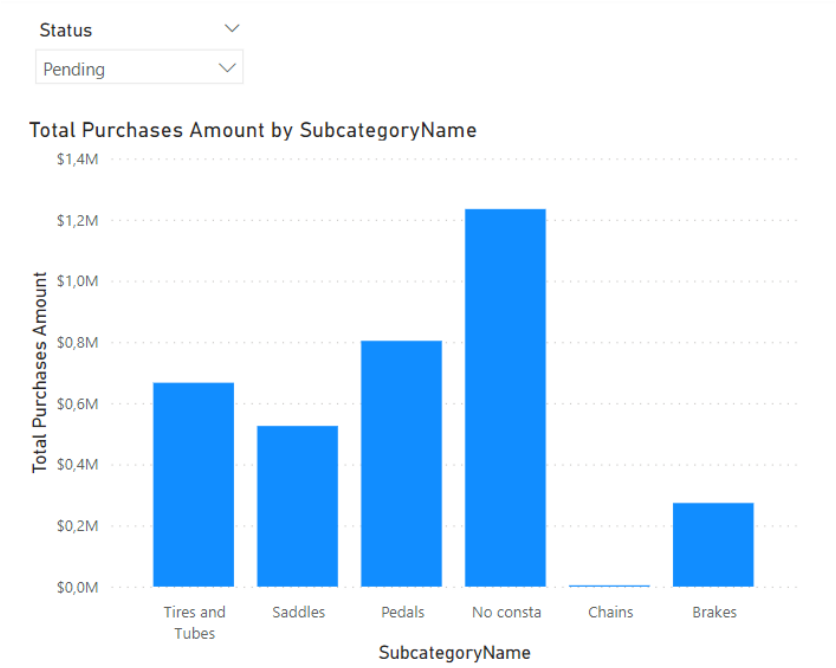


Desarrollo: Vamos a ver varios casos en los que aplicaremos este tip. El primero consiste en hacer que el rango de un único gráfico no varíe en función de lo que se seleccione en la segmentación.

En primer lugar, una vez tengamos el gráfico diseñado (en nuestro caso, un gráfico de barras), añadimos una segmentación sobre el lienzo:



Como se puede observar, si no hay ningún valor seleccionado, el valor máximo del eje Y es 20 millones, pero al seleccionar algún valor de la segmentación, el valor máximo del eje pasa a ser 1.4 millones.

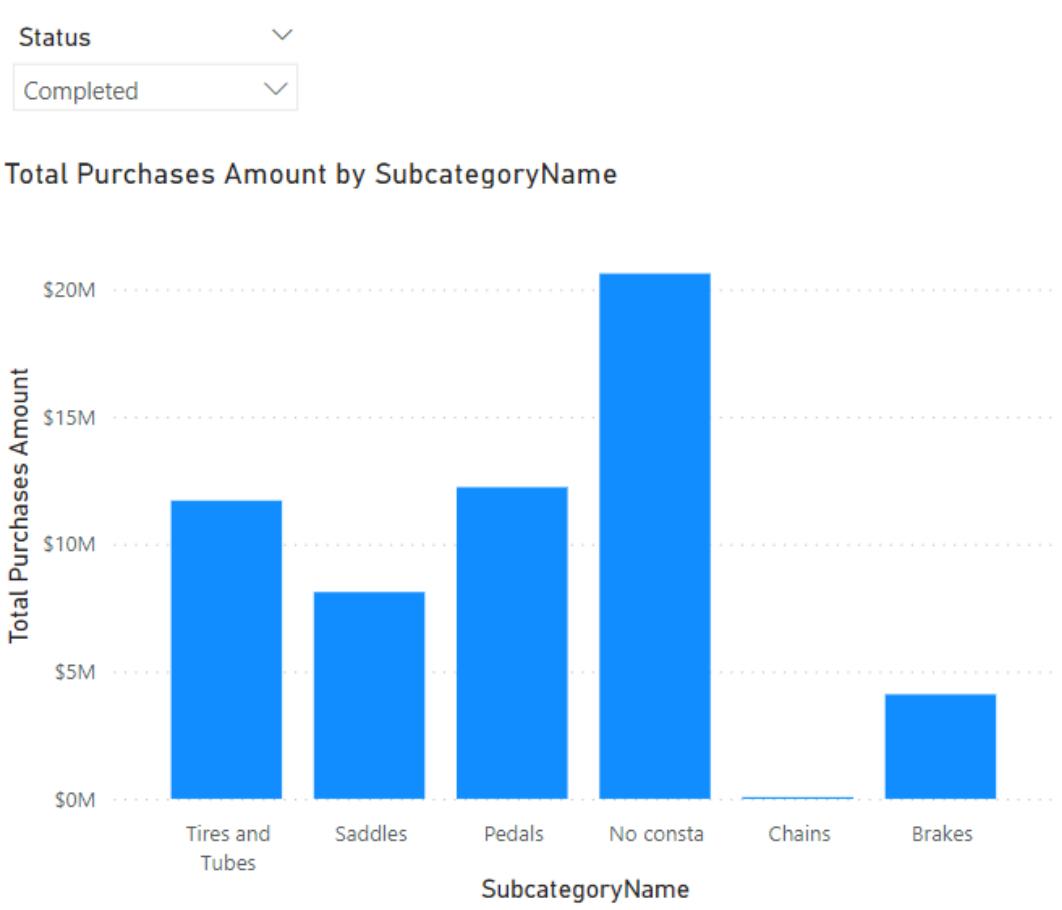


Para mantener el rango del eje Y fijo, vamos a crear una medida que utilizaremos como valor máximo del eje Y:

```
Y-Axis range subcategories =
    CALCULATE (
        ROUNDUP (
            MAXX (
                VALUES ( Dim_product[SubcategoryName] ),
                [Total Purchases Amount]
            ),
            -6
        ),
        REMOVEFILTERS ( Dim_status[Description] )
    )
```



Ahora el rango del eje Y se mantiene al seleccionar un valor en la segmentación:



En el segundo caso, replicaremos la misma lógica pero esta vez, el objetivo no es que el rango no varíe sino que el rango sea el mismo en los dos gráficos para poder compararlos.

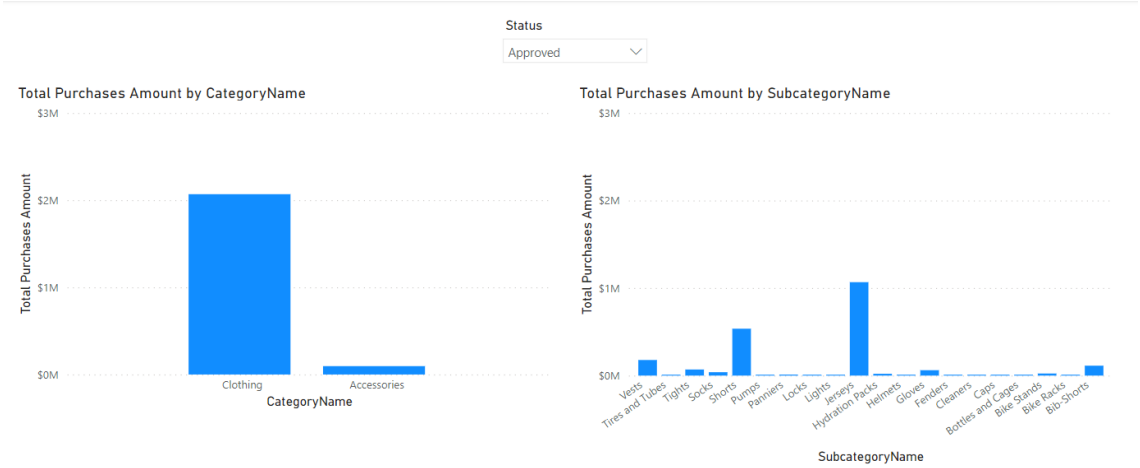


Como vemos, en el primer gráfico el valor máximo es 30 millones y en el segundo, 20, por lo que no se pueden hacer comparaciones directas.

En este caso, tendremos que crear otra medida con 2 variables (una por cada eje) y como resultado, se calcula el valor máximo de las mismas:

```
Y axis range Category and Subcategory =
    VAR MaxRangeByCategory =
        ROUNDUP (
            MAXX (
                VALUES ( Dim_product[CategoryName] ),
                [Total Purchases Amount]
            ),
            -6
        )
    VAR MaxRangeBySubcategory =
        ROUNDUP (
            MAXX (
                VALUES ( Dim_product[SubcategoryName] ),
                [Total Purchases Amount]
            ),
            -6
        )
    VAR Result = MAX ( MaxRangeByCategory, MaxRangeBySubcategory )
    RETURN Result
```

Como en el caso anterior, utilizaremos esta nueva medida como valor máximo del eje Y:



Aunque el rango vaya variando según el valor por el que se filtra, siempre se mantiene el mismo rango en ambos ejes.

12. Calcular porcentajes precisos con RLS

El objetivo de este tip es mostrar cómo calcular correctamente los porcentajes cuando hay algunos datos ocultos al estar aplicada RLS.



Desarrollo: Pueden existir escenarios en los que necesitemos aplicar seguridad por filas, pero en los que se quiera calcular porcentajes o cualquier agregación en función de los datos totales y no solo de los que están filtrados. En este tip se especifica cómo hacer estos cálculos.

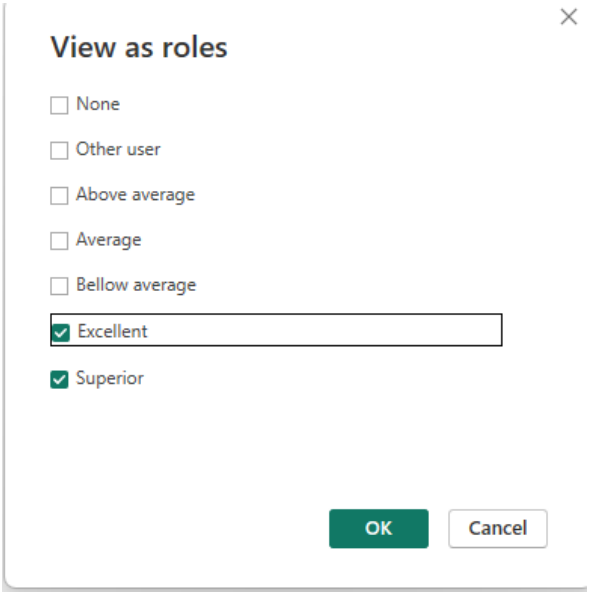
En primer lugar, creamos una medida del porcentaje de compras sobre el total:

```
% over All =
DIVIDE (
    [Total Purchases Amount],
    CALCULATE (
        [Total Purchases Amount],
        ALL ( Dim_vendor )
    )
)
```

Los porcentajes calculados sin utilizar RLS son:

Rating	Purchases Amount	% over All
Above average	$2.530.562	3,97 %
Average	$2.347.422	3,68 %
Below average	$1.857.171	2,91 %
Excellent	$5.562.933	8,72 %
Superior	$51.493.907	80,72 %
Total	**$63.791.995**	**100,00 %**

Pero si necesitamos añadir RLS para que los vendedores de cada categoría solo puedan ver los datos de su categoría, los porcentajes varían. Por ejemplo, al aplicar RLS con los filtros de “Superior” y “Excellent”, los porcentajes se ven así:

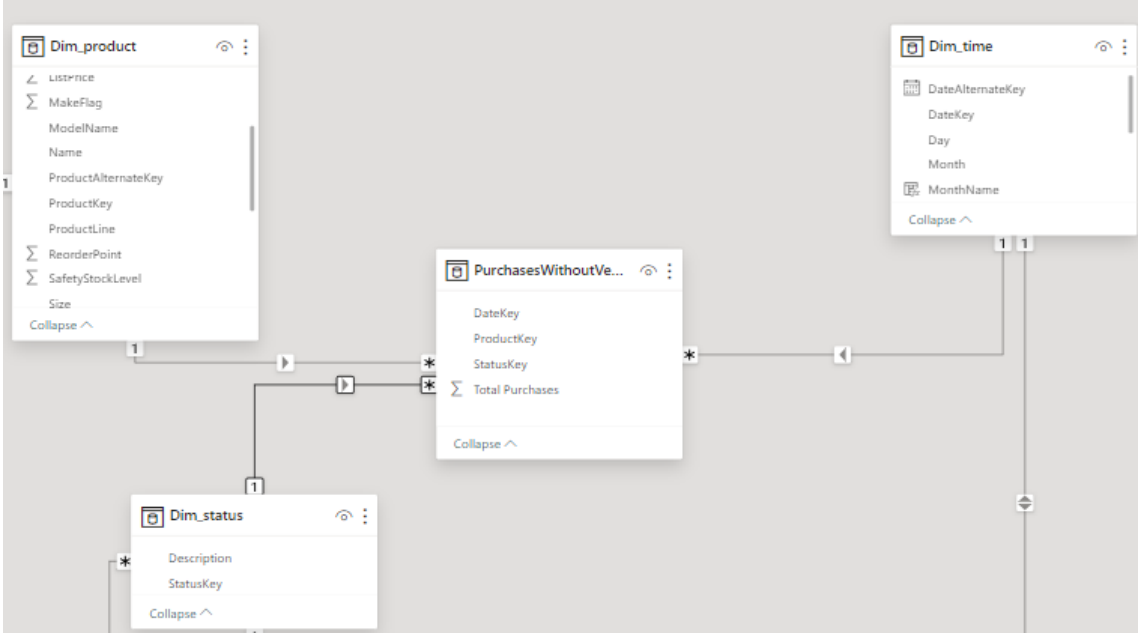


Rating	Purchases Amount	% over All
Excellent	$5.562.933	9,75 %
Superior	$51.493.907	90,25 %
Total	**$57.056.840**	**100,00 %**

Ahora los porcentajes se calculan en función de los datos filtrados, por eso han cambiado. Pero si el requisito es mostrar los porcentajes en función del total, aunque esté aplicado RLS, tenemos que hacer algunos cambios.

En primer lugar, creamos una tabla calculada y las relaciones necesarias en el modelo:

```
PurchasesWithoutVendor =
ADDCOLUMNS (
    SUMMARIZE (
        Fact_purchases,
        'Dim_product'[ProductKey],
        Dim_time[DateKey],
        Dim_status[StatusKey]
    ),
    "Total Purchases", [Total Purchases Amount]
)
```



A continuación, creamos la siguiente medida para calcular los porcentajes sobre la tabla nueva:

```
% Not Secured =
DIVIDE (
    [Total Purchases Amount],
    SUM ( PurchasesWithoutVendor[Total Purchases] )
)
```

Finalmente, al volver a aplicar RLS, vemos que se muestran los porcentajes correctos, aunque las categorías de vendedores están filtradas:

Rating	Purchases Amount	% over All	% Not Secured
Excellent	$5.562.933	9,75 %	8,72 %
Superior	$51.493.907	90,25 %	80,72 %
Total	**$57.056.840**	**100,00 %**	**89,44 %**

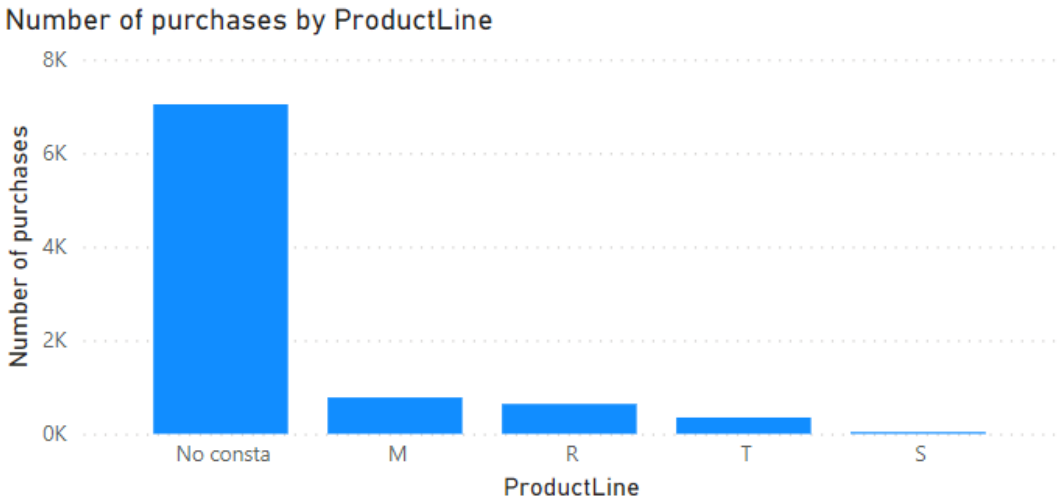
3. VISUALIZACIÓN

13. Formato condicional basado en campos de tipo String

El objetivo de este apartado es mostrar cómo se puede aplicar el formato condicional basado en campos de tipo String.

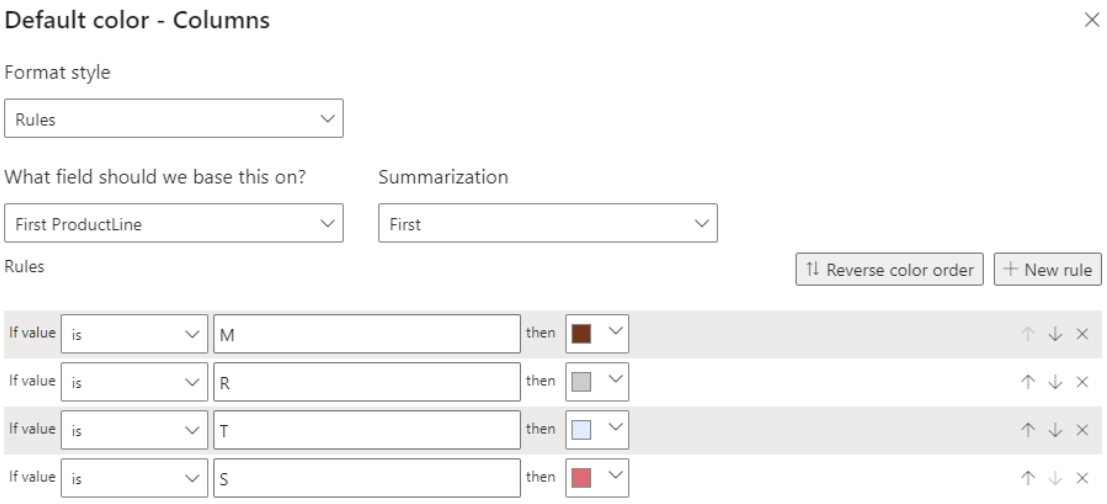


Desarrollo: Este tip se puede aplicar a cualquier tipo de gráfico pero en nuestro caso, lo haremos sobre un diagrama de barras.

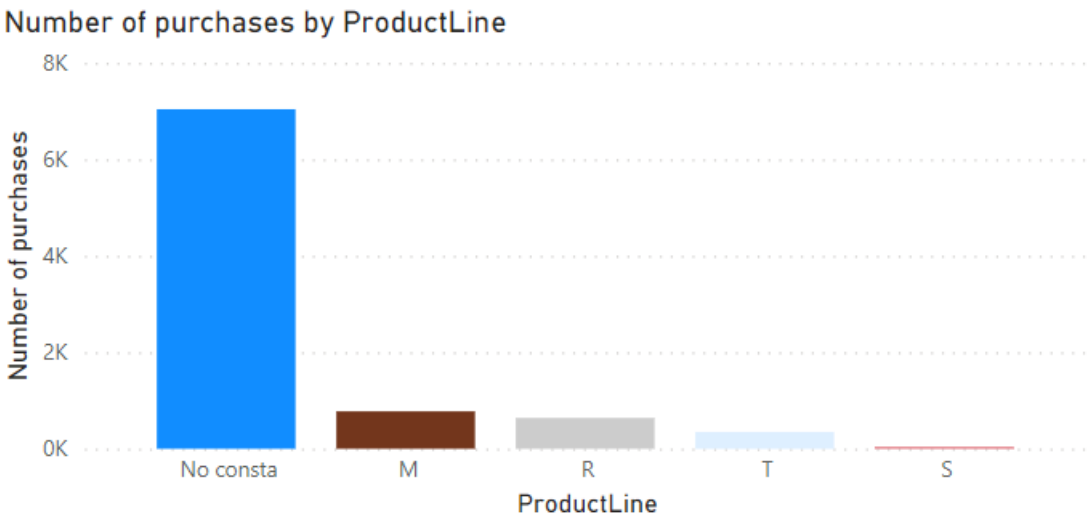


Lo primero que hay que hacer, es abrir el panel de formato, sección Colores/Columnas.

En la ventana que se abre, se tiene que seleccionar Rules en el campo Format Style y se selecciona el campo por el que vamos a basar el código de colores. A continuación, añadimos las reglas que se consideren necesarias añadiendo las condiciones adecuadas y seleccionando el color. En el caso de los valores que no se incluyan en estas condiciones, se muestran con el color por defecto que tiene el visual.



El resultado es el siguiente:



Como el valor “No consta” no está en ninguna de las condiciones, se mantiene el color que tenía por defecto.

14. Etiquetas de datos personalizadas con medidas

El objetivo de este apartado es mostrar un ejemplo de cómo personalizar las etiquetas de datos añadiendo un icono dependiendo de si el valor ha aumentado o disminuido con respecto al año anterior.



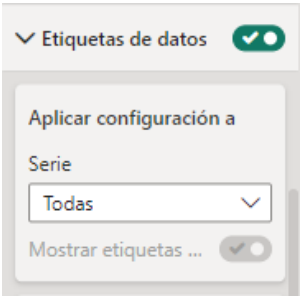
Desarrollo: Para ello, en primer lugar, vamos a crear una nueva medida en el modelo llamada Total Purchases measure label:

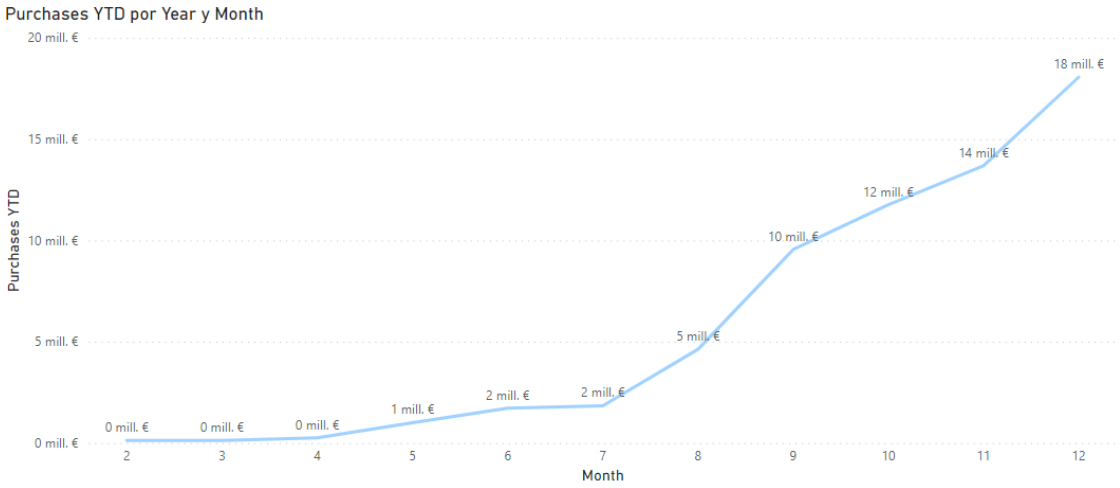
```
Total Purchases Measure Label =
var _thisyearpurchases = [Purchases YTD]
var _lastyearpurchases = [Purchases YTD-1]
var _up = "🟩"
var _down = "🟥"
var _result =
SWITCH(TRUE(),
    _thisyearpurchases >= _lastyearpurchases, _up,
    _thisyearpurchases < _lastyearpurchases, _down
)
return
FORMAT([Purchases YTD], "Currency") & _result
```

Para abrir el panel de iconos, utilizar las teclas Windows y "." y seleccionar el icono deseado. En este caso, se ha elegido el cuadrado verde para indicar los casos en los que las compras del año seleccionado sean mayores que las del año anterior, y el cuadrado rojo para el caso contrario.

De esta forma, de un vistazo rápido se puede hacer un balance general de cómo ha ido el año con respecto al anterior.

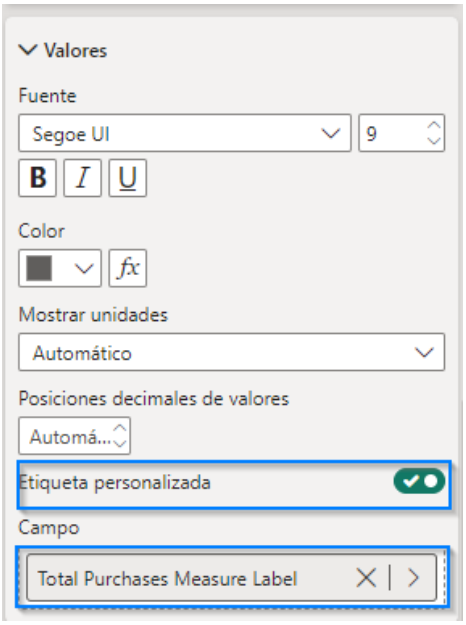
A continuación, creamos un gráfico de líneas para ver la evolución por año y mes y activamos las etiquetas de datos en el panel de formato:



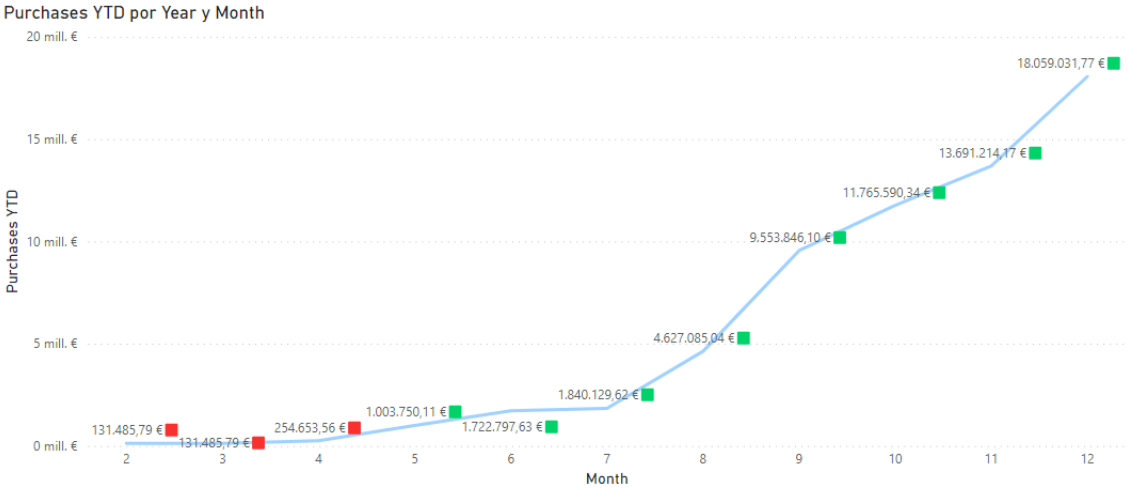


Para añadir la medida que hemos creado previamente como etiqueta de datos, en el panel de formato, si desplegamos la opción de etiquetas de datos y hacemos scroll hacia abajo, vemos que hay una sección llamada “valores”.

Desplegando esta sección, tenemos que activar la opción de “Etiqueta personalizada” y en la casilla “Campo”, añadir la medida creada.



El resultado final sería el siguiente:



15. Usos del formato dinámico

El objetivo de este apartado es mostrar diferentes usos del formato dinámico.



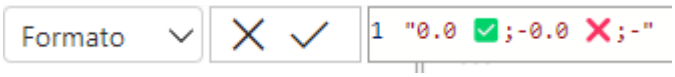
Desarrollo: El primer uso consiste en crear una medida con formato dinámico y dependiendo de si el número es positivo o negativo, añadir un icono al lado.

Para ello, lo primero que tenemos que hacer es crear la medida, en nuestro caso:

```
Sales Amount Dynamic = SUM(FactInternetSales[SalesAmount])
```

A continuación, cambiamos el formato a "Dinámico" y veremos en la barra de fórmulas el formato de la medida. La primera parte antes del ";" representa los valores positivos, la segunda, los negativos y la última, los valores nulos.

En nuestro caso, vamos a añadir un tick verde si el número representado en la medida es positivo y una cruz roja si es negativo.



Para añadir un emoticono, tenemos que presionar a la vez la tecla Windows y ".".



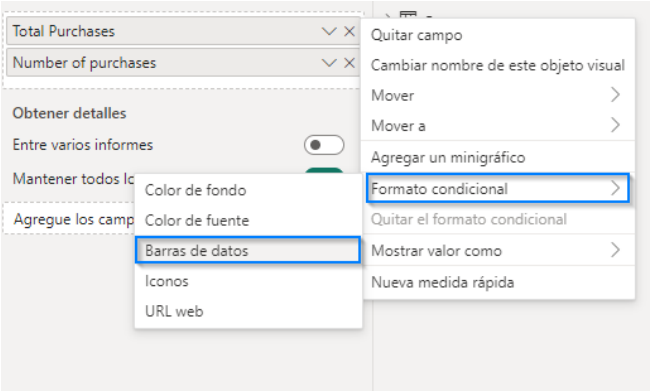
16. Barras de datos en tablas y columnas

El objetivo de este apartado es mostrar cómo se pueden representar en forma de barras los datos de una tabla.

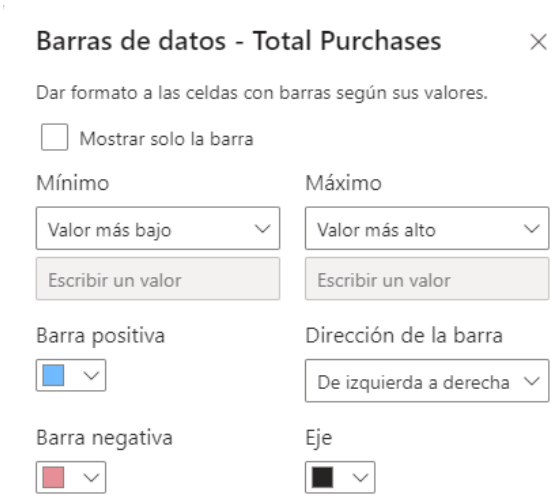


Desarrollo: En primer lugar, creamos una matriz con al menos dos niveles y las medidas que se consideren. Este caso es interesante realizarlo con dos niveles como mínimo porque la funcionalidad principal es comparar los diferentes valores y si solo hay un nivel, se puede ordenar el eje de forma ascendente o descendente. Sin embargo, si hay varios niveles y se ordena el eje, los valores quedan ordenados, pero dentro de cada categoría, por tanto, es más útil verlos en forma de barra.

Existen dos formas para activar esta funcionalidad. La primera es haciendo click derecho sobre la medida en la que se quiere aplicar, seleccionar la opción de formato condicional -> barras de datos.

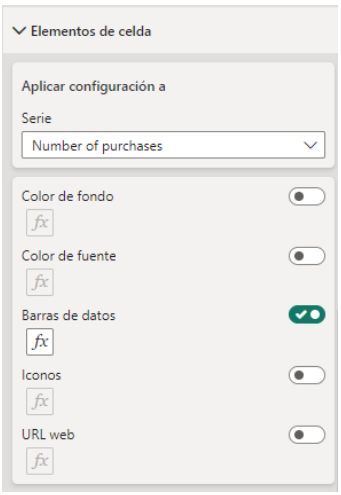


A continuación, aparecerá la siguiente ventana en la que se pueden configurar diferentes aspectos de las barras como los colores, valores mínimo y máximo, etc:



Es importante elegir colores que contrasten con el color de los valores para que, si se superpone la barra con el valor numérico, este pueda seguir siendo legible.

La segunda forma es activar esta funcionalidad es desde el panel de formato, columna específica, seleccionar la medida y activar las barras de datos.



El resultado final es:

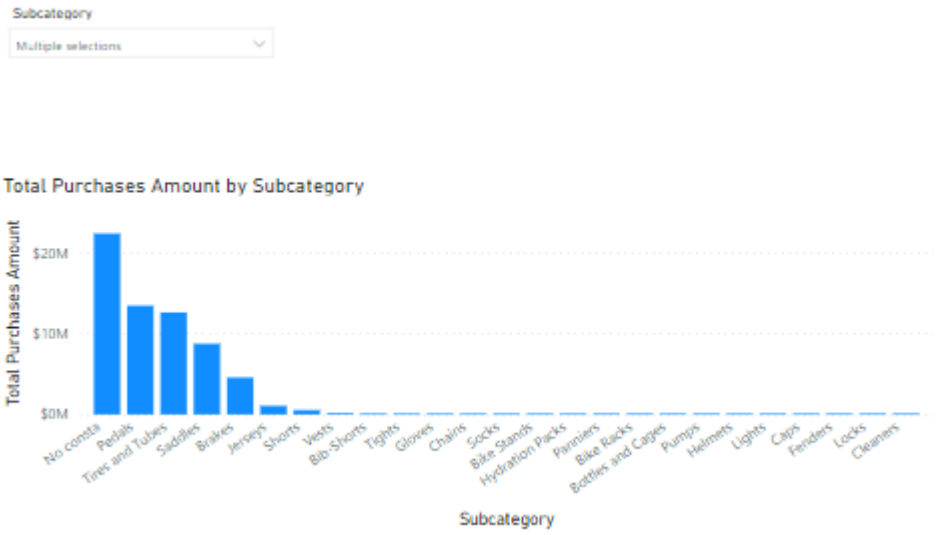
CategoryName	Total Purchases	Number of purchases
Accessories	**$12.671.936**	**722**
Clothing	**$2.068.471**	**38**
Components	**$26.742.023**	**1193**
Brakes	**$4.555.898**	**100**
Front Brakes	$2.277.949	50
Rear Brakes	$2.277.949	50
Chains	**$47.219**	**50**
Chain	$47.219	50
Pedals	**$13.413.431**	**528**
HL Mountain Pedal	$1.766.855	51
HL Road Pedal	$1.455.057	42
LL Mountain Pedal	$1.766.561	102
LL Road Pedal	$1.558.730	90
ML Mountain Pedal	$2.709.041	102
ML Road Pedal	$2.390.330	90
Touring Pedal	$1.766.855	51
Saddles	**$8.725.476**	**515**
HL Mountain Seat/Saddle 2	$1.598.687	71
HL Road Seat/Saddle 2	$1.373.520	61
HL Touring Seat/Saddle	$878.152	39
LL Mountain Seat/Saddle 2	$707.720	61
LL Road Seat/Saddle 1	$823.740	71
LL Touring Seat/Saddle	$464.079	40
ML Mountain Seat/Saddle 2	$1.021.245	61
ML Road Seat/Saddle 2	$1.188.662	71
ML Touring Seat/Saddle	$669.669	40
Total	**$41.482.429**	**1953**

17. Resaltar datos dinámicamente según las segmentaciones

El objetivo de este tip es mostrar cómo resaltar los datos de un visual dinámicamente dependiendo de la selección que se haga en la segmentación.



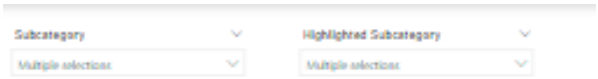
Desarrollo: En primer lugar, añadimos al lienzo un objeto visual (en nuestro caso, un gráfico de barras) y una segmentación por la que filtrar los datos y deshabilitamos la interacción entre el filtro y el gráfico.

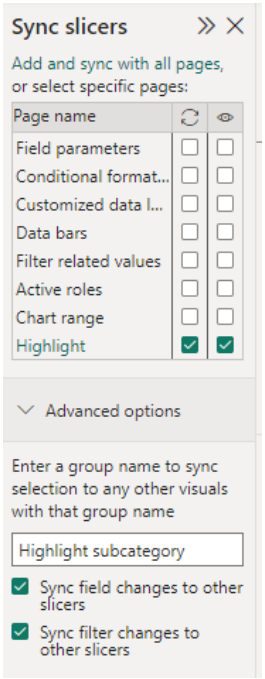


A continuación, creamos una tabla auxiliar que únicamente tenga un campo con los valores de las diferentes subcategorías. Esta tabla queda aislada del modelo (sin relacionar con ninguna otra).

```
1  Highlight Table = VALUES(Dim_product[SubcategoryName])
```

Después, añadimos al lienzo otra segmentación con este nuevo campo y agrupamos ambas en el mismo grupo (Highlighted subcategory) para que así, al seleccionar los valores en una de ellas, también queden seleccionados en la otra.





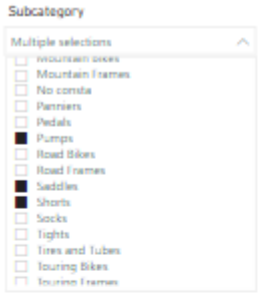
El siguiente paso es crear una medida que vamos a utilizar para aplicar el formato condicional. Opcionalmente podemos añadir al lienzo una tarjeta con esta medida para ver el número de valores seleccionados en la segmentación.

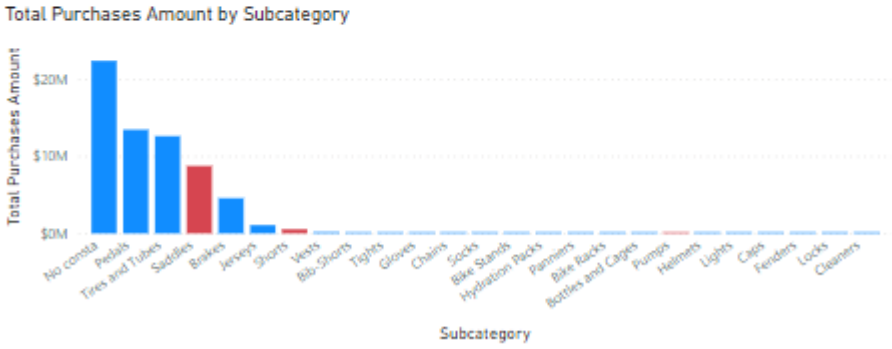
```
Conditional Formatting =
VAR SelectedSubcategory = COUNTROWS(
                    INTERSECT(
                        VALUES('Highlight Table'[SubcategoryName])
                        , VALUES(Dim_product[SubcategoryName])
                        )
                        )
RETURN
IF(
    ISFILTERED('Highlight Table'[SubcategoryName] )
        , SelectedSubcategory
        , 0
        )
```

Aplicamos el formato condicional al gráfico utilizando esta medida.



Por último, ocultamos el filtro de la tabla auxiliar, ya que el que vamos a utilizar es el filtro de la dimensión.





Como podemos ver, hay 3 valores seleccionados en la segmentación que se corresponden con las 3 barras de diferente color del gráfico.

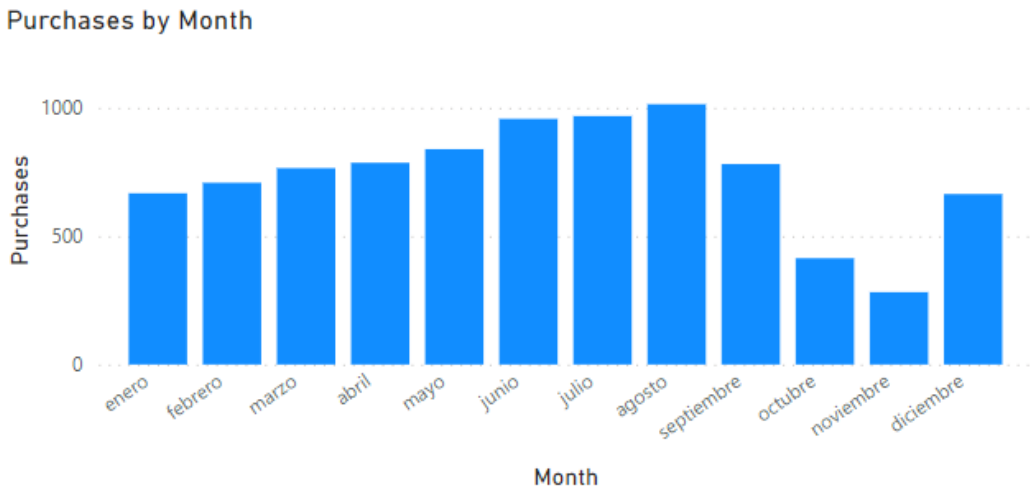
Si queremos añadir otras segmentaciones pero que no afecten al cambio de colores, debemos dejarlas fuera del grupo de segmentaciones que hemos creado.

18. Resaltar valores mínimo y máximo de un gráfico

El objetivo de este tip es mostrar cómo resaltar dinámicamente los valores mínimo y máximo de un gráfico para poder identificarlos más rápidamente.



Desarrollo: Este tip tiene sentido en los casos en los que el eje horizontal no esté ordenado ascendente o descendentemente sino siguiendo otro criterio, por ejemplo, el temporal. En este caso, partimos de un gráfico de barras con un eje temporal.



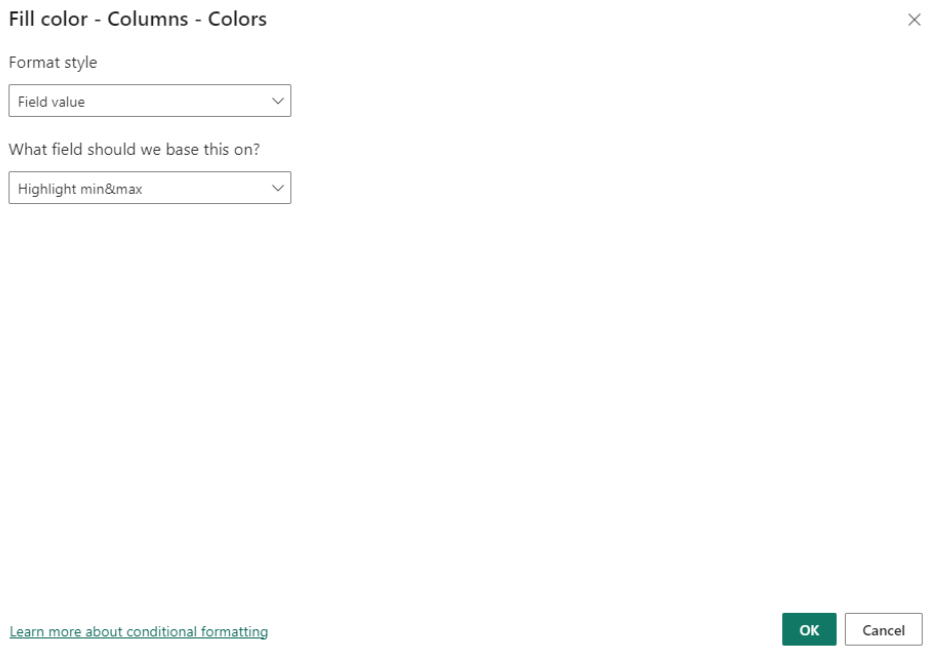
En primer lugar, creamos una nueva medida donde se calculan los valores mínimo y máximo:

```
Highlight min&max =
var maxPurchases =
    MAXX(
        ALL('Calendar'[Month name], 'Calendar'[Month]),
        [Number of purchases])
 var minPurchases =
    MINX(
        ALL('Calendar'[Month name], 'Calendar'[Month]),
        [Number of purchases])
 return maxPurchases
```

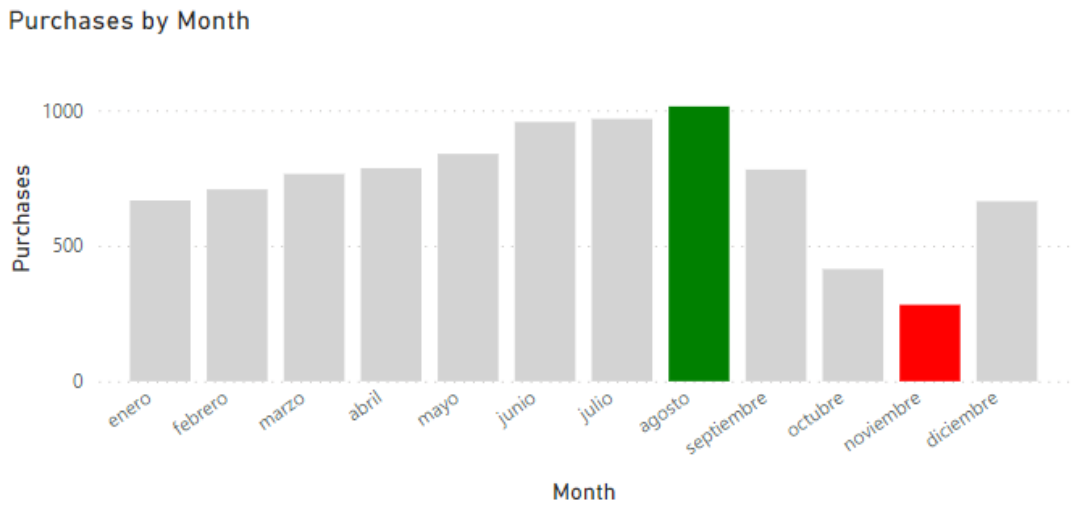
Modificamos la medida anterior añadiendo otra nueva variable llamada “Color”.

```
Highlight min&max =
var maxPurchases =
    MAXX(
        ALL('Calendar'[Month name], 'Calendar'[Month]),
        [Number of purchases])
 var minPurchases =
    MINX(
        ALL('Calendar'[Month name], 'Calendar'[Month]),
        [Number of purchases])
 var color =
        SWITCH(
            TRUE(),
            [Number of purchases] = maxPurchases, "Green",
            [Number of purchases] = minPurchases, "Red",
            "Light Grey")
 return color
```

Aplicamos el formato condicional sobre el gráfico de barras utilizando esta medida:



Obteniendo como resultado final el gráfico con las dos barras de mínimo y máximo resaltadas sobre las demás:



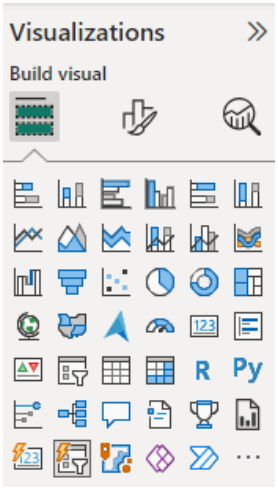
19. Personalización de segmentaciones

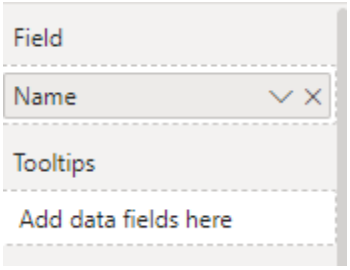
El objetivo de este tip es mostrar cómo se pueden personalizar las segmentaciones.



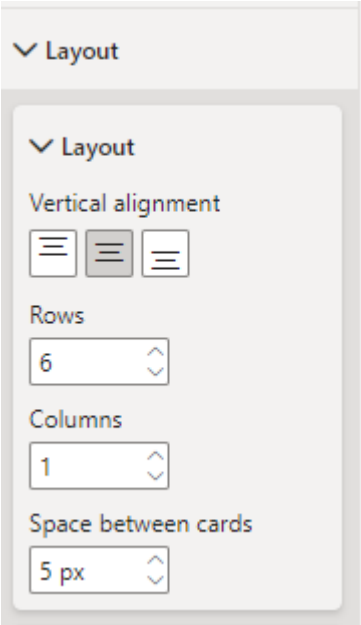
Desarrollo: Este tip tiene como objetivo mostrar cómo se puede personalizar los objetos de segmentaciones.

En primer lugar, vamos a añadir al lienzo un objeto de tipo segmentación y vamos a añadir un único campo.

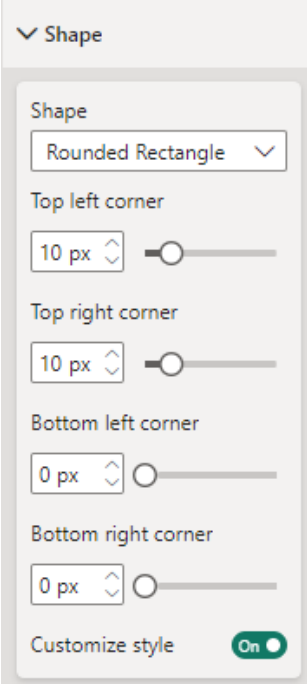


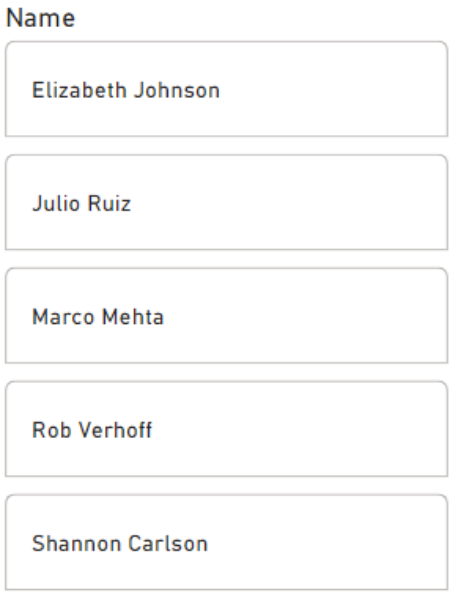


En el apartado de Layout podemos configurar el número de filas y columnas de la segmentación, así como la alineación vertical y el espacio entre los botones.

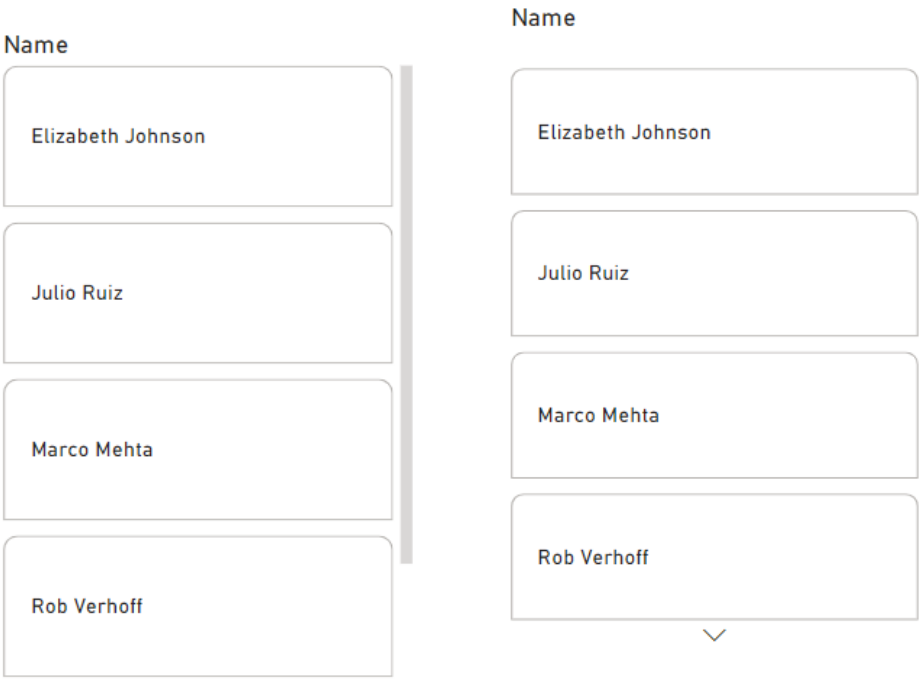


En Shape, si activamos la opción de Customize style, podemos personalizar cada una de las esquinas del botón. En este caso, vamos a personalizarlas de forma que las superiores sean redondeadas y las inferiores, rectas.

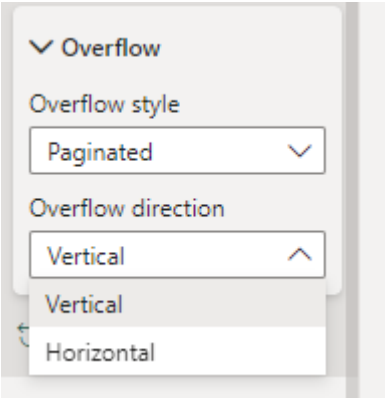



Name
Elizabeth Johnson
Julio Ruiz
Marco Mehta
Rob Verhoff
Shannon Carlson

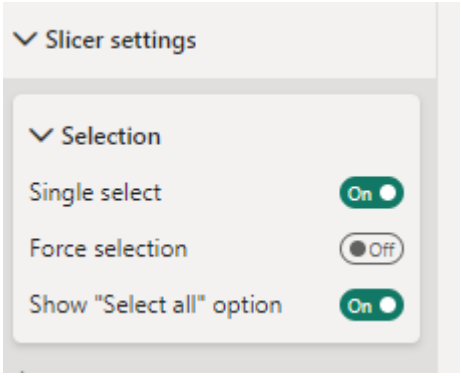
En el caso de que haya más valores que las filas que se han establecido para mostrar, esto se puede personalizar en la sección Layout->Overflow, existiendo dos opciones: scroll continuo o paginado.



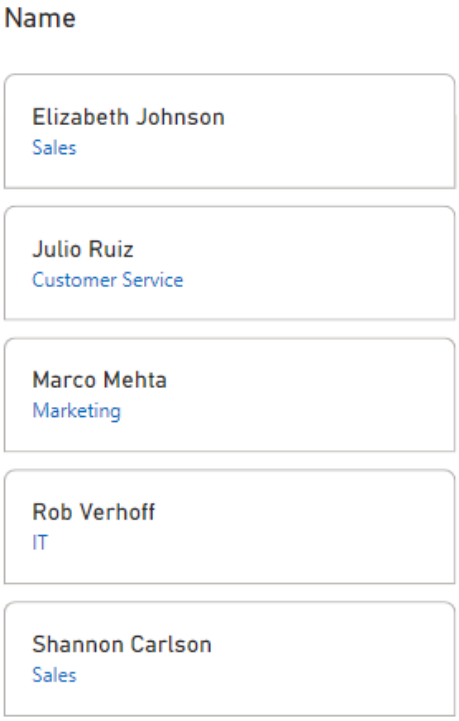
En ambos casos, se puede elegir dirección vertical u horizontal.

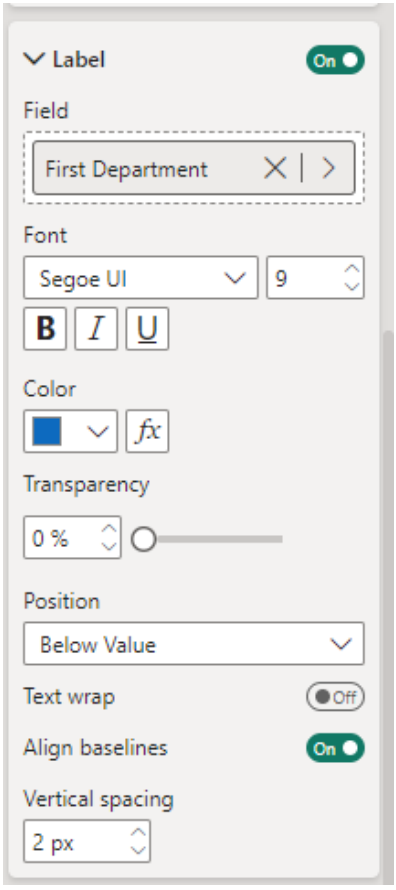


También se puede añadir la opción de Seleccionar todo o de forzar selección:



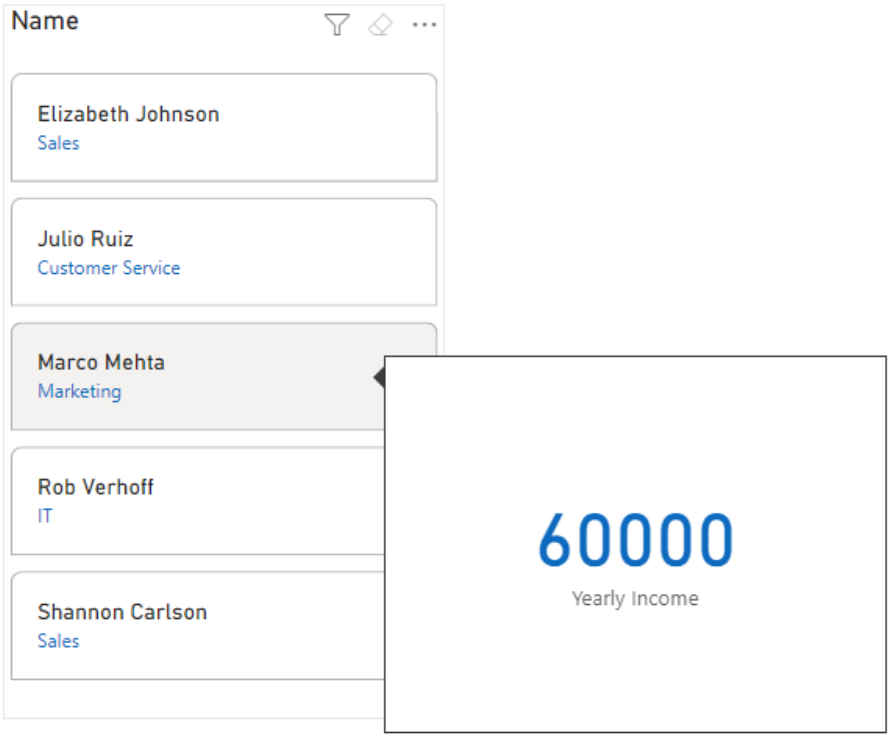
Además, podemos añadir otro campo o medida en Label para proporcionar más información de cada una de las opciones.





Además, se pueden añadir imágenes en cada botón y personalizar los diferentes estados (por defecto, al pasar por encima del botón, al presionar el botón y botón seleccionado).

También se pueden agregar tooltips desde General->Tooltips.



4. OTROS

20. Tips&Tricks sobre jerarquías en Scorecards

El propósito de este tip es recoger algunas recomendaciones que reducen los tiempos de carga cuando se utilizan jerarquías.



- Habilitar el almacenamiento en caché de consultas

  Habilitar el almacenamiento en caché de datasets conectados que se encuentren en áreas de trabajo Premium puede acelerar la carga cacheando los resultados de consultas anteriores.

  Nota: los datasets deben estar en áreas de trabajo Premium para habilitar esta característica.

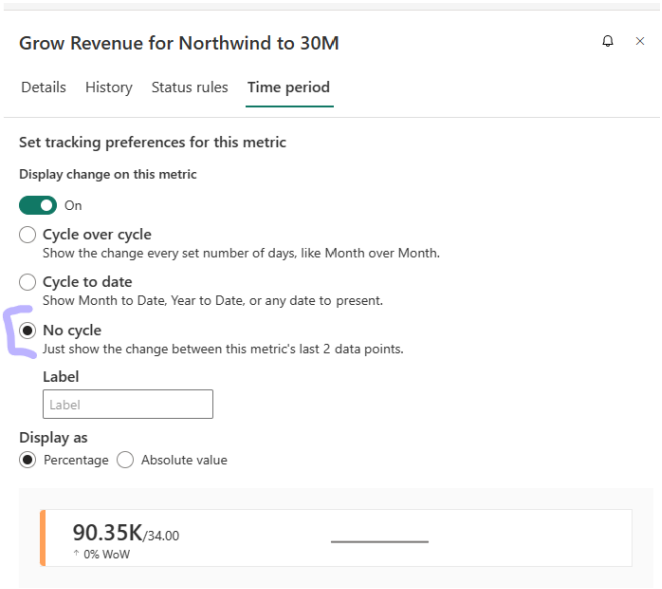
- Crear filtros de fecha relativa

  Utilizar filtros de fecha relativa en visuales que estén relacionados puede mejorar el rendimiento de la carga limitando el número de datapoints y de consultas que se necesitan ejecutar por cada selección a cargar de la jerarquía.

- Conectarse a datos de visuales drilled down/ expanded

  Creando una jerarquía de fecha en el objeto visual que se necesite y profundizando a la granularidad de fecha que se necesite, también mejora los tiempos de carga.

- Ajustar la periodicidad de las métricas a “Ningún ciclo” o a un período de tiempo menor



- Conectarse a un datapoint único en lugar de a una serie temporal

Si no se necesita el historial completo, se puede conectar a un datapoint concreto y considerar el historial desde ese punto.

21. Leer roles de seguridad activos en Power BI

El propósito de este tip es enseñar cómo se pueden mostrar los roles activos de un usuario, incluso en el caso de que este tenga asociado más de uno.



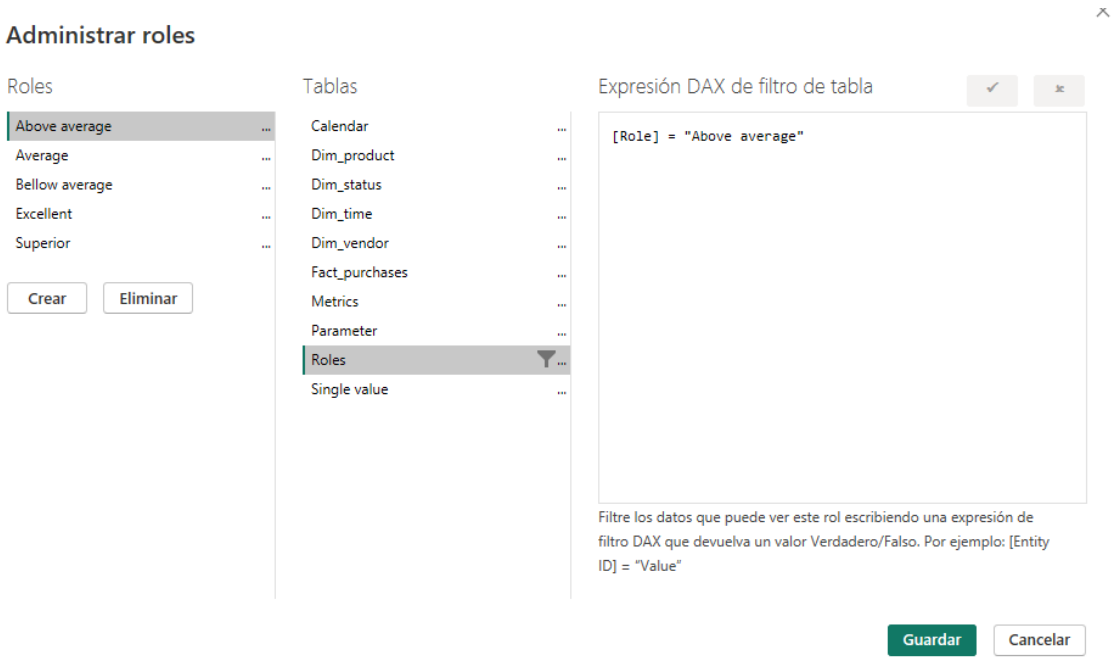
Desarrollo: En primer lugar, necesitamos una tabla a la que llamaremos “Roles”, que contiene una única columna llamada “Role” cuyos valores son los siguientes:

```
1 Roles =
2 SELECTCOLUMNS (
3     { "Superior", "Excellent", "Average", "Above average", "Below average" },
4     "Role", [Value]
5 )
```

Role
Superior
Excellent
Average
Above average
Below average

Esta tabla queda aislada en el modelo, es decir, no está relacionada con ninguna otra.

El siguiente paso es crear cada uno de los roles correspondientes en la tabla “Roles” de la siguiente forma:



En el caso de que un usuario sólo tenga asociado un rol, se podría utilizar la siguiente medida:

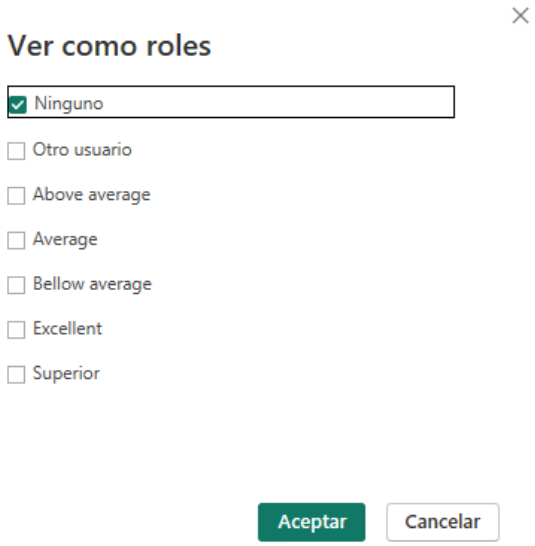
```
Active Role =
SELECTEDVALUE ( Roles[Role] )
```

Pero esta medida devuelve el valor (En blanco) en el caso de que el usuario tenga más de un rol asociado. En estos casos, podemos utilizar la siguiente medida:

```
Active Roles =
CONCATENATEX (
    VALUES ( Roles[Role] ),
    Roles[Role],
    ", "
)
```

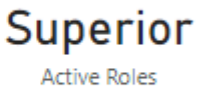
Esta medida devuelve los valores correctos tanto si el usuario tiene un único rol como si tiene varios.

Podemos simular el comportamiento de estas medidas con la opción “Ver como” de la pestaña “Modelado”.



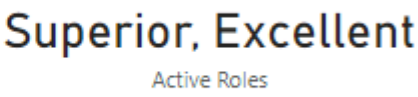
En el primer caso, seleccionamos solo un rol y vemos que ambas medidas muestran el valor correcto:





Pero si seleccionamos más de un rol:

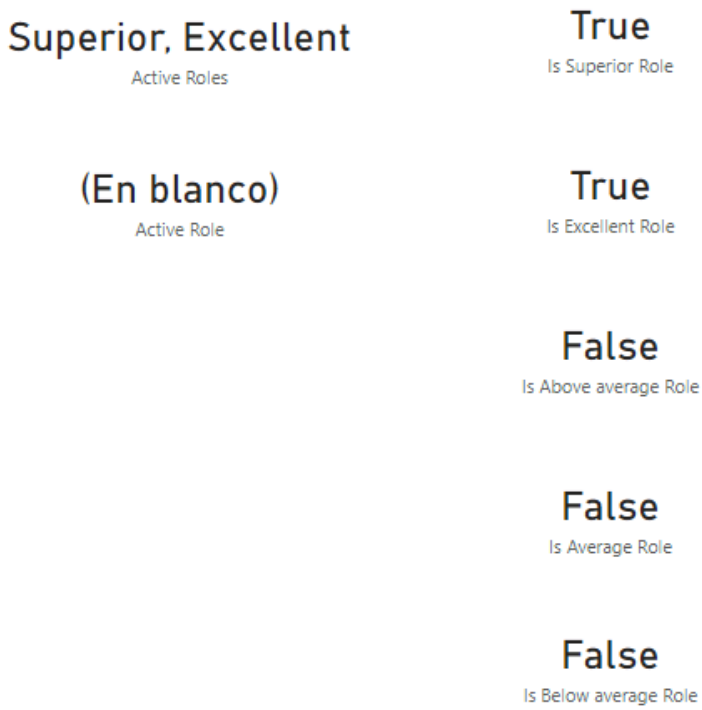




En el caso de que queramos comprobar si un rol está activo o no, podemos crear la siguiente medida, una por cada rol que tengamos:

```
1 Is Superior Role =
2 "Superior" IN VALUES ( Roles[Role] )
```

Y el resultado sería el siguiente:



22. Alternativa a Sort by column

El propósito de este tip es enseñar cómo ordenar los valores de un campo sin utilizar la función sort by.



Desarrollo: En nuestro caso, se quieren ordenar los valores de los meses de la tabla de calendario. Como podemos observar, nuestra tabla de calendario tiene los meses del año ordenados alfabéticamente, pero queremos ordenarlos en orden natural.

MonthName	Purchases Amount
abril	$5.292.199
agosto	$7.155.397
diciembre	$4.667.058
enero	$4.834.159
febrero	$4.744.827
julio	$8.002.157
junio	$6.681.067
marzo	$6.040.789
mayo	$6.409.615
noviembre	$1.925.624
octubre	$2.778.029
septiembre	$5.261.074
Total	**$63.791.995**

Lo que vamos a hacer es modificar el campo MonthName de la siguiente manera, de forma que se añade un número diferente de caracteres unichar dependiendo del mes:

```
MonthName =
REPT(UNICHAR(8203), 12-MONTH(Dim_time[DateAlternateKey])) &
FORMAT('Dim_time'[DateAlternateKey], "MMMM")
```

De esta forma, quedan los meses ordenados siguiendo el orden temporal:

MonthName	Purchases Amount
enero	$4.834.159
febrero	$4.744.827
marzo	$6.040.789
abril	$5.292.199
mayo	$6.409.615
junio	$6.681.067
julio	$8.002.157
agosto	$7.155.397
septiembre	$5.261.074
octubre	$2.778.029
noviembre	$1.925.624
diciembre	$4.667.058
Total	**$63.791.995**

23. Replicar funciones SUMIF y COUNTIF en Power BI

El propósito de este tip es enseñar cómo implementar en Power BI un comportamiento similar al de las funciones SUMIF y COUNTIF de Excel.



Desarrollo: Para los casos en los que se requiera una funcionalidad parecida al de las funciones de Excel SUMIF y COUNTIF, este tip muestra la alternativa en Power BI.

Para la función SUMIF, vamos a crear la siguiente columna en la tabla de hechos:

```
Date SUMIF =
VAR currentdate = Fact_purchases[OrderDate]
return
    SUMX(
        FILTER(
            Fact_purchases,
            Fact_purchases[OrderDate] = currentdate
        ),
        Fact_purchases[OrderQuantity]
    )
```

Para el caso de la función COUNTIF, vamos a partir de la fórmula anterior, pero haciendo unas pequeñas modificaciones:

```
Date COUNTIF =
VAR currentdate = Fact_purchases[OrderDate]
return
    COUNTROWS(
        FILTER(
            Fact_purchases,
            Fact_purchases[OrderDate] = currentdate
        )
    )
```

Los resultados son los siguientes:

VendorKey	StatusKey	OrderQuantity	ReceivedQuantity	RejectedQuantity	UnitPrice	LineTotal	EmployeeTitle	OrderDateKey	ShipDateKey	DueDateKey	OrderDate	ShipDate	DueDate	Date COUNTIF	Date SUMIF
45	4	4	3	0	50,26	201,04	Buyer	20110416	20110425	20110430	16/04/2011	25/04/2011	30/04/2011	5	563
3	1	3	3	0	45,58	136,74	Buyer	20110416	20110425	20110430	16/04/2011	25/04/2011	30/04/2011	5	563
2	4	550	550	0	16,09	8847,30	Buyer	20110416	20110425	20110430	16/04/2011	25/04/2011	30/04/2011	5	563
80	3	3	2	1	57,03	171,08	Purchasing Assistant	20110416	20110425	20110430	16/04/2011	25/04/2011	30/04/2011	5	563
3	1	3	3	0	45,12	135,36	Buyer	20110416	20110425	20110430	16/04/2011	25/04/2011	30/04/2011	5	563
87	4	550	468	0	26,60	14628,08	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
63	4	3	3	0	47,47	142,41	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
63	4	3	3	0	45,37	136,11	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
63	4	3	3	0	49,64	148,93	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
63	4	3	3	0	45,37	136,11	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
63	4	3	3	0	43,27	129,81	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
94	4	550	550	0	27,06	14882,18	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
94	4	550	550	0	33,58	18468,45	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
82	4	550	550	0	37,09	20397,30	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
94	4	550	550	0	46,06	25334,93	Buyer	20110430	20110509	20110514	30/04/2011	09/05/2011	14/05/2011	10	2765
57	4	60	60	0	26,37	1581,93	Buyer	20111214	20111223	20111228	14/12/2011	23/12/2011	28/12/2011	19	718
56	4	60	60	0	25,42	1525,23	Purchasing Manager	20111214	20111223	20111228	14/12/2011	23/12/2011	28/12/2011	19	718
100	3	3	3	3	48,76	146,29	Purchasing Assistant	20111214	20111223	20111228	14/12/2011	23/12/2011	28/12/2011	19	718

24. Alternativa del servicio de Power BI a las tablas pivotantes de Excel

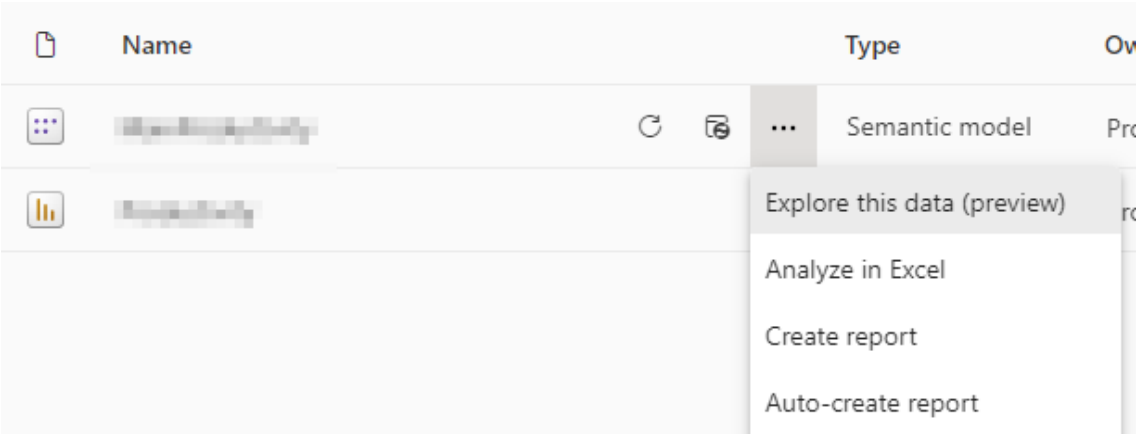
El objetivo de este tip es enseñar cómo se puede replicar un comportamiento similar al de las tablas pivotantes de Excel sin necesidad de tener que exportar los datos.



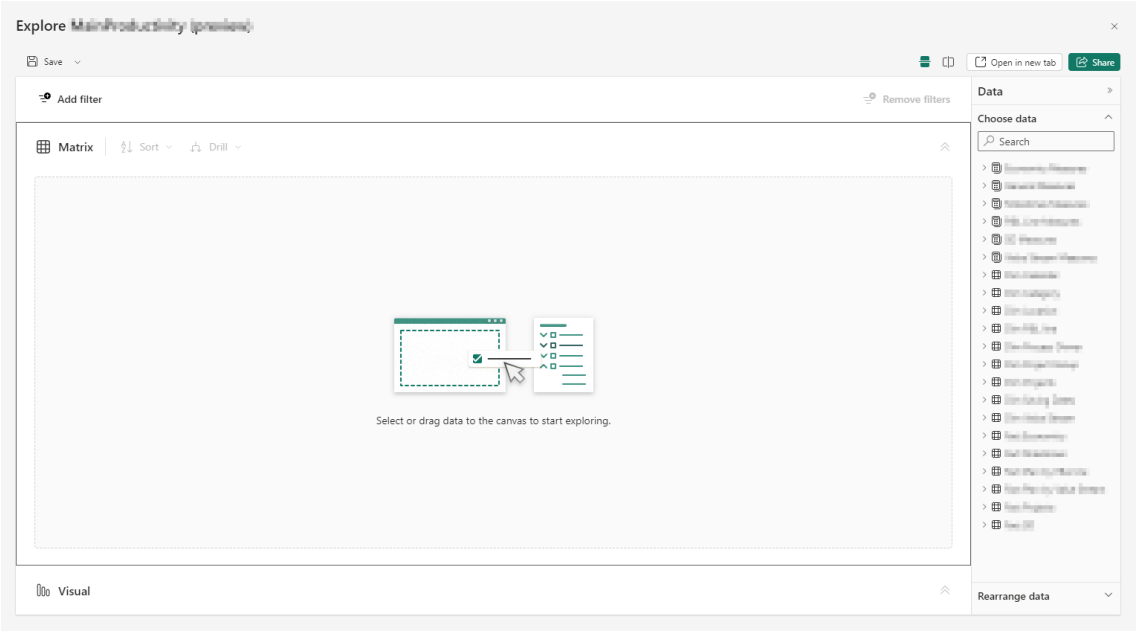
Desarrollo: Hay veces en las que después de publicar el dataset y el informe en el servicio de Power BI, algunos usuarios acostumbrados a Excel quieren hacer un análisis diferente sobre el conjunto de datos y exportan los datos a un libro de Excel para poder crear tablas pivotantes.

Esto ya no es necesario, ya que desde el propio servicio de Power BI se puede hacer dicho análisis, incluso con alguna funcionalidad añadida.

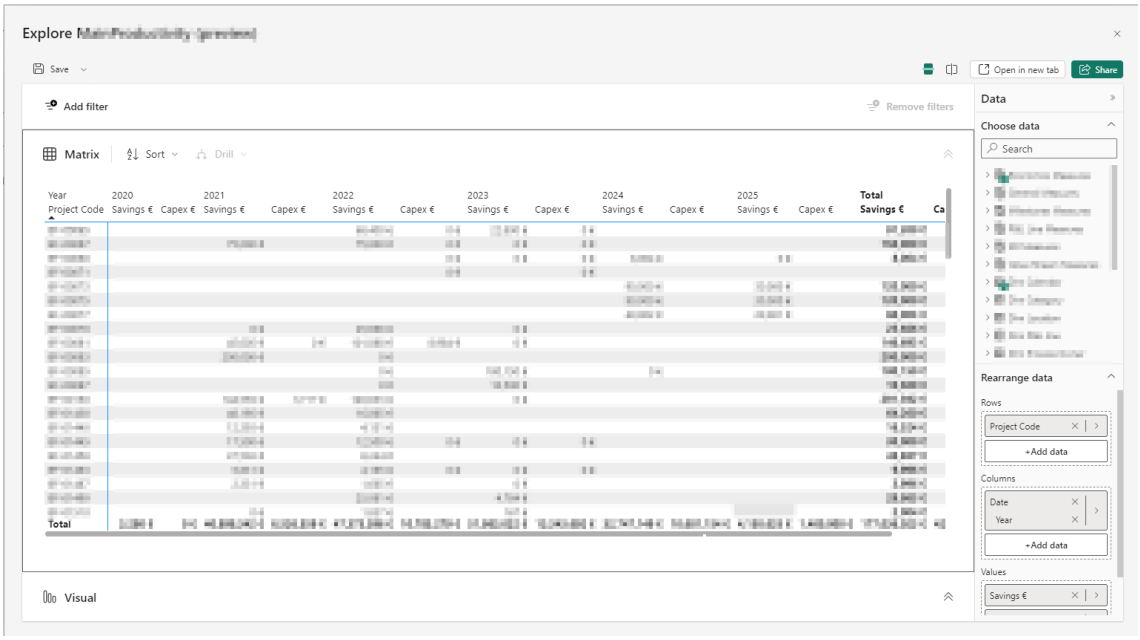
Lo único que debemos hacer es acceder l servicio de Power BI, y al área de trabajo donde esté publicado el dataset sobre el que queremos hacer un análisis más detallado. Una vez localizado el dataset, hacemos click sobre los tres puntos que aparecen y seleccionamos la opción de “Explore this data”.



A continuación, se nos abrirá una ventana flotante en la que podemos comenzar a añadir los campos que necesitemos en una matriz.

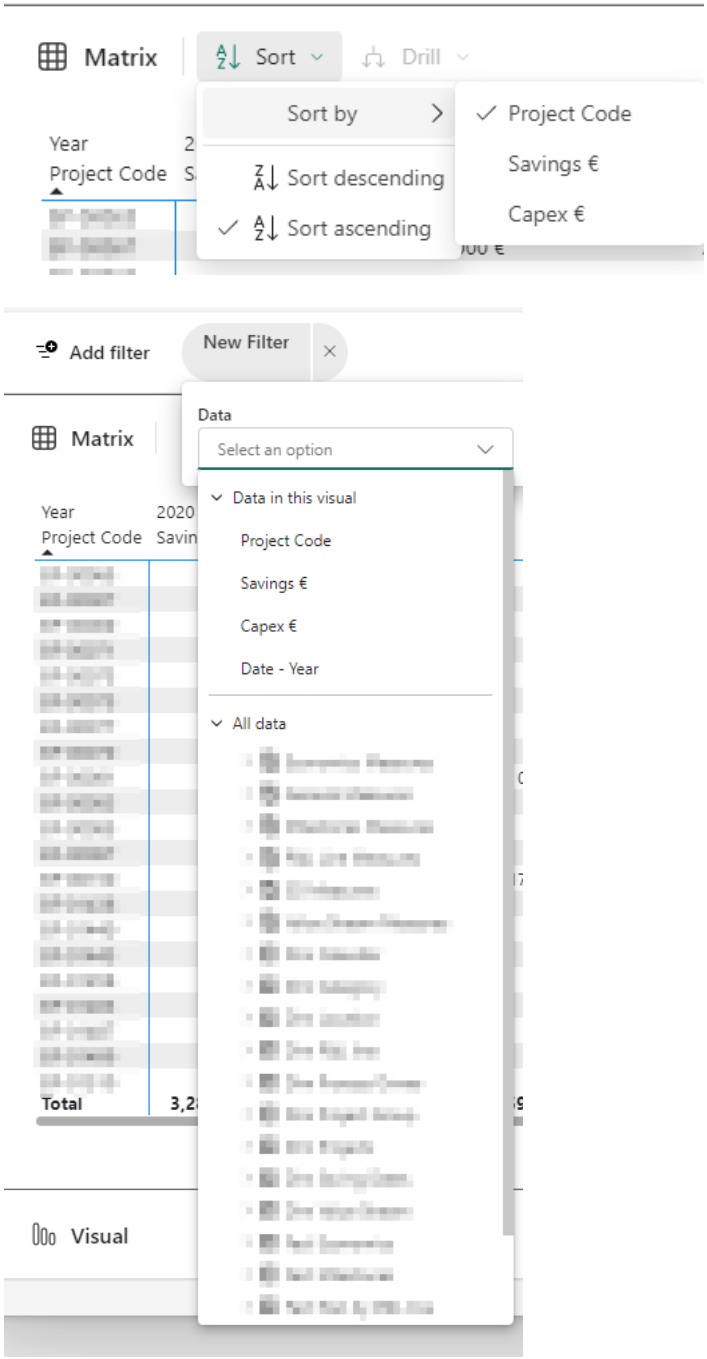


En nuestro caso, hemos añadido los siguientes campos:

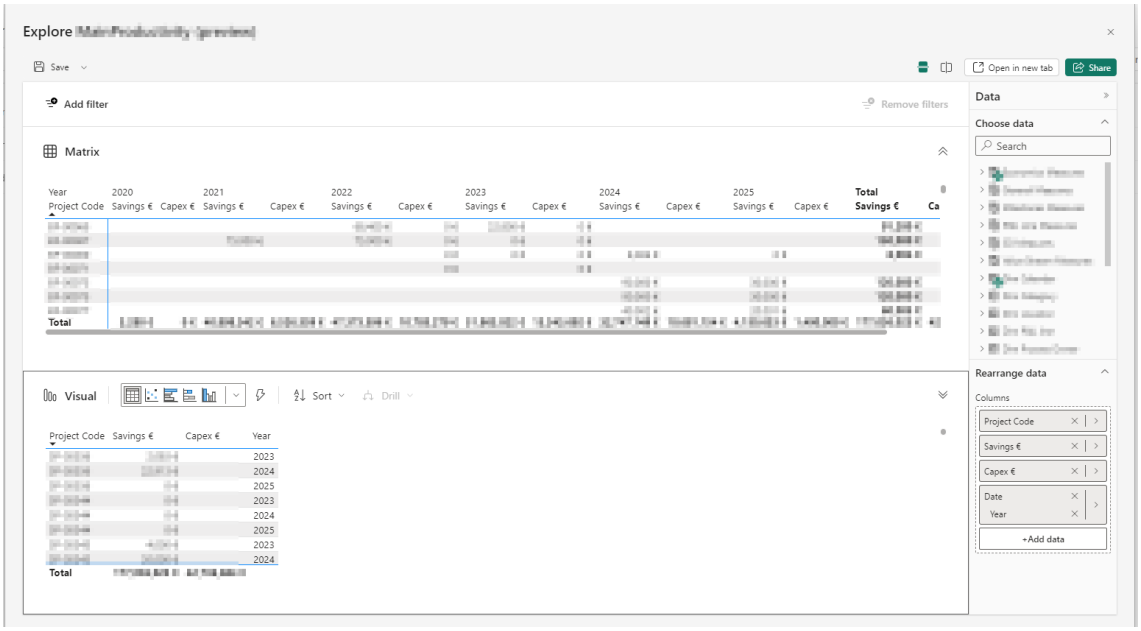


Si fuera necesario, también se pueden crear agregaciones sobre los campos añadidos en la matriz.

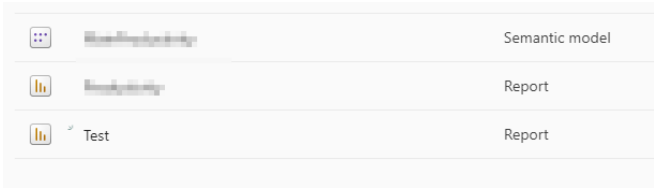
Además, podemos añadir los filtros que sean necesarios y ordenar los datos.



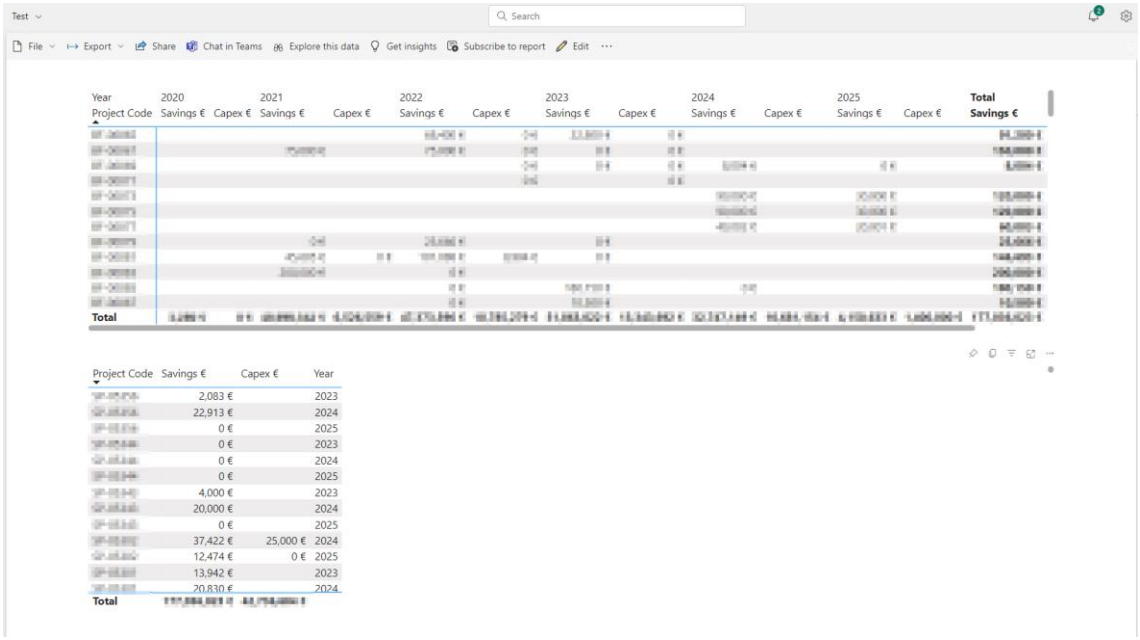
Como funcionalidad adicional, podemos generar un visual de cualquiera de los tipos disponibles en Power BI al mismo tiempo que vemos la matriz.



Cuando hayamos finalizado el análisis, podemos guardarlo como una exploración (esta funcionalidad solo está disponible para áreas de trabajo con capacidad Premium) o guardarlo como un informe (disponible para todas las áreas de trabajo). Solo tenemos que añadir un nombre y seleccionar el área de trabajo donde lo queremos guardar. También tenemos la opción de compartirlo.



Al guardarlo como un informe, tenemos todas las opciones disponibles para un informe normal.



5. MAS TRUCOS, VIDEOTUTORIALES Y PAPERS

1. Integracion SAP - PowerBI
2. PowerBI Trucos (Vol I)
3. PowerBI Trucos (Vol II)
4. PowerBI + Synapse Analytics (paso a paso)
5. 30 Consejos y Buenas Prácticas para hacer un proyecto de Power BI con éxito
6. Cómo crear diseños de Dashboards espectaculares con PowerBI
7. Videotutorial: Trabajando con Python en Power BI
8. Aplicación PowerBi Turismo
9. Aplicación PowerBI Financiera I
10. Aplicación PowerBI Financiera II
11. Aplicación PowerBI eCommerce
12. Aplicación PowerBI Salud
13. Aplicación PowerBi Smart City
14. Aplicación PowerBI Energía
15. Aplicación PowerBI Sports Analytcis
16. Power BI Premium Utilization and Metrics
17. PowerBI Embedded: Funcionamiento y costes
18. Bravo para PowerBI
19. Como integrar Power BI con Microsoft Dynamicssalesfo
20. SQL Server Profiler para Power BI
21. Como usar Report Analyzer en PowerBI, para mejorar el rendimiento
22. Power BI embebido en Jupyter Notebook
23. Tabular Editor para Power BI: Videotutorial y manual en español
24. Personaliza tus gráficas en Power BI con Charticulator y Deneb
25. Comparativa PowerBI vs Amazon QuickSight
26. Como usar emoticonos en PowerBI
27. Buenas prácticas con Dataflows en Power BI
28. Power Automate para Power BI: Cómo funciona
29. ALM Toolkit para Power BI
30. Os presentamos Goals in Power BI para hacer Scorecards
31. Tutorial gratuito en español sobre Power BI Report Builder
32. Conoce PowerBI Diagram View (Visual Data Prep). Paso a paso
33. Futbol Analytics, lo que hay que saber
34. Dashboard de medicion de la calidad del aire en Madrid
35. Como funciona Microsoft Power BI? Videoturial de Introducción
36. Big Data para PowerBI
37. Quieres crear aplicaciones empresariales usando PowerBI, PowerApps y Power Automate de forma conjunta?
38. Power BI tip: Uso de parámetros what-if
39. Como integrar Salesforce y PowerBI

40. Videotutorial: Usando R para Machine Learning con PowerBI
41. Las 50 claves para aprender y conocer PowerBI
42. PowerBI: Arquitectura End to End
43. Usando Python con PowerBI
44. PowerBI + Open Source = Sports Analytics
45. Comparativa de herramientas Business Intelligence
46. Use Case Big Data "Dashboards with Hadoop and Power BI"
47. Todas las presentaciones del Workshop 'El Business Intelligence del Futuro'
48. Descarga Paper gratuito: Zero to beautiful (Data visualization)
49. SAP connection tools for process automation: Microsoft, Pentaho, Talend (User Guide)
50. PowerBI Tips (Vol. 3)
51. Enmascaramiento de datos en PowerBI con Azure
52. Mejores KPIs para Dashboards de PowerBI en Turismo
53. Planificación, análisis del rendimiento y escalado de un cluster de Puerta de enlace en PowerBI
54. Cómo actualizar tablas en PowerBI
55. Cómo usar ChatGPT con PowerBI
56. Report Bursting con PowerBI
57. 20 KPIs financieros para usar en Dashboards con PowerBI
58. Refresco incremental en PowerBI

stratebi
Analytics and Big Data

Microsoft
Gold Partner