

# Tarea S6.01. Análisis avanzado con visualizaciones interactivas de Power BI

Radostin Pavlov

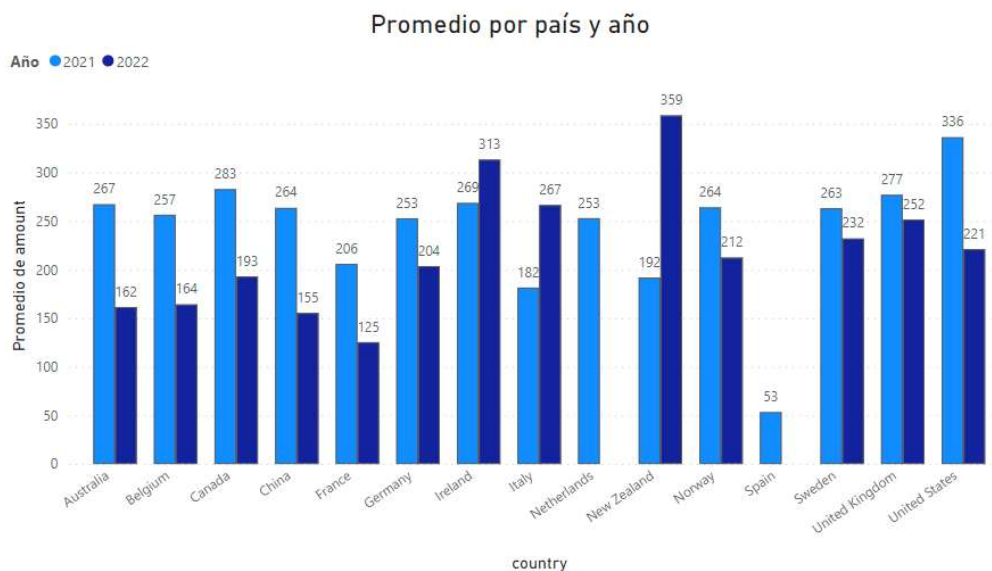
## Nivel 1

### Ejercicio 1

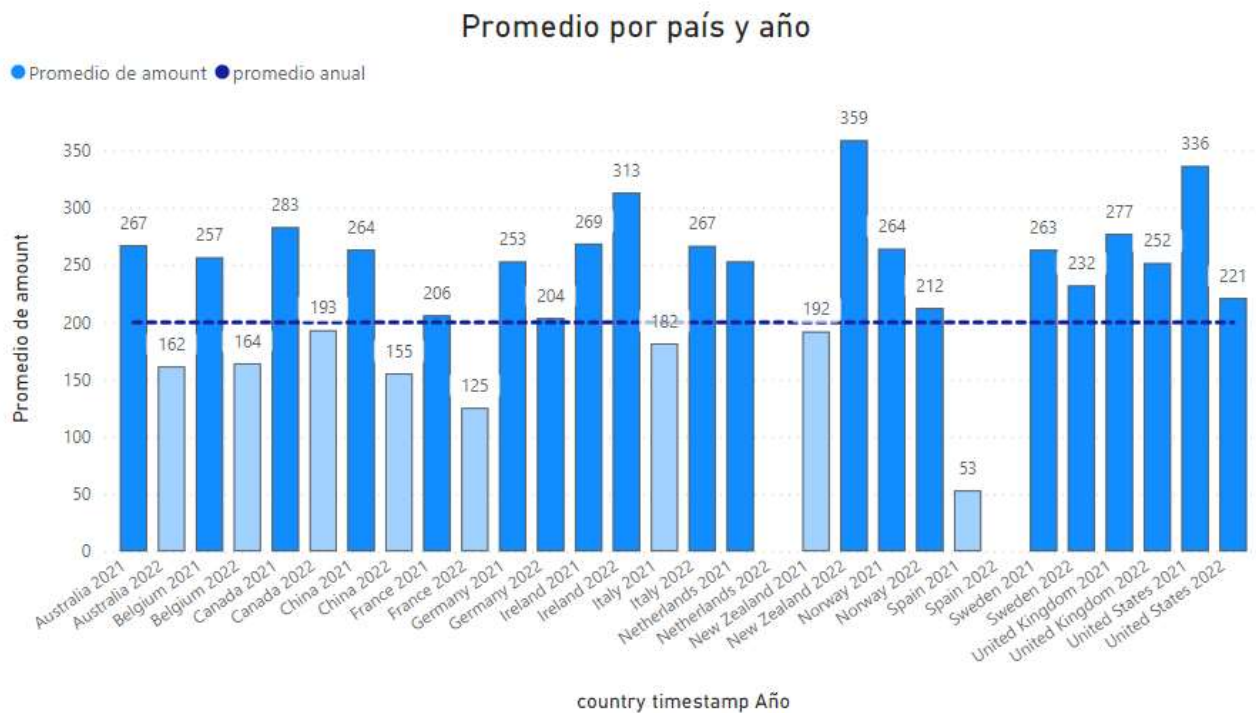
La empresa necesita evaluar el rendimiento de las ventas a nivel internacional. Como parte de este proceso, te piden que elijas un gráfico en el que se detalle la media de ventas desglosadas por país y año en una misma presentación visual. Es necesario señalar las medias menores de 200 euros anuales.

### Solución

Lo ideal sería poder crear un gráfico de columnas agrupadas como el siguiente y después establecer formato condicional para cambiar el color de las barras de valores inferiores al objetivo, pero no se ha encontrado la posibilidad de hacerlo.



Como solución alternativa se ha creado un gráfico de columnas agrupadas y líneas, que permite destacar el valor objetivo, así como aplicar color condicional:

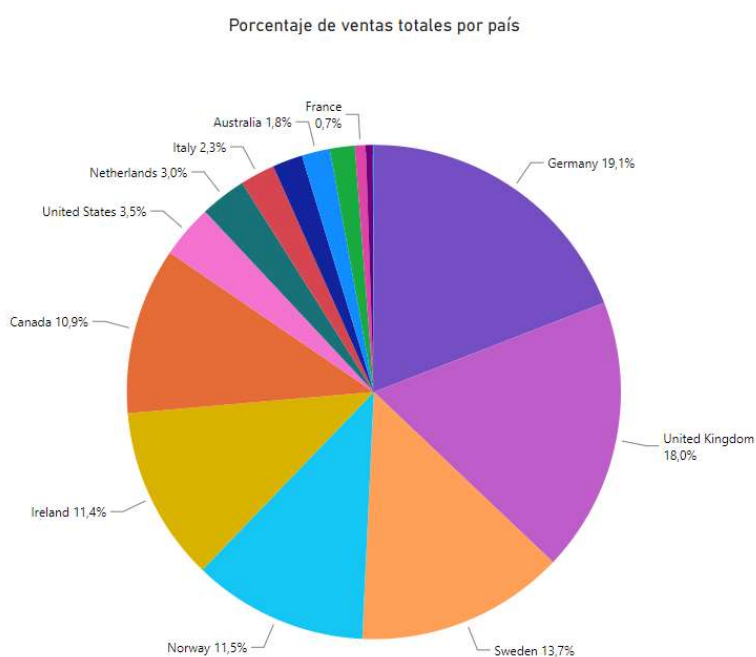


## - Ejercicio 2

La empresa está interesada en obtener una visión general de las transacciones realizadas por país y desea identificar una métrica clave para evaluar ese aspecto del negocio. Tu tarea es crear una visualización en la que se identifique el porcentaje de las ventas totales de la empresa que representa a cada país respecto al total global.

### Solución:

Un gráfico de tipo circular (“tarta”) es idóneo para representar porcentajes:



### - Ejercicio 3

Diseña un indicador visual en Power BI para analizar la diferencia de ventas entre 2022 y 2021 en cada país. La empresa está interesada en comprender cómo han variado las ventas en diferentes países durante este período y desea identificar cualquier disminución o aumento significativo en las ventas.

### Solución

Una posibilidad es emplear de nuevo un gráfico de barras agrupadas, que da una idea cualitativa sobre la diferencia:



Si se desea cuantificar la diferencia en ventas por país , conviene crear una tabla nueva mediante Power Query, DAX o SQL o con la diferencia y a continuación representarla en una gráfico de barras.

Procedimiento con Power Query:

- Se duplica la tabla transactions ( = Origen{[Schema="ventas",Item="transactions"]} [Data] )
- Se expande companies y se selecciona el campo country  
( = Table.ExpandRecordColumn(ventas\_transactions, "ventas.companies", {"country"}, {"ventas.companies.country"}) )
- Se crea una columna nueva extrayendo el año del timestamp

```
( = Table.AddColumn("#Se expandió ventas.companies", "Personalizado", each Date.Year([timestamp])) )
```

- Se quitan todas las columnas menos la del año, la del país y la de la cantidad

```
( = Table.SelectColumns("#Columnas con nombre cambiado",{ "Año", "ventas.companies.country", "amount"}) )
```

- La tabla se duplica
- Una de las tablas se filtra por Año = 2021 y la otra por Año = 2022
- Cada una de las dos tablas se agrupa por el campo country empleando sum([amount]) como función de agregación:

```
= Table.Group("#Filas filtradas", {"ventas.companies.country"}, {{"Recuento", each List.Sum([amount]), type nullable number}})
```

- Se genera una tabla nueva en base a un full join externo de las dos tablas usando el campo común país

```
( = Table.FuzzyNestedJoin("#Año 2021", {"ventas.companies.country"}, "#Año 2022", {"ventas.companies.country"}, "Año 2022", JoinKind.FullOuter, [IgnoreCase=true, IgnoreSpace=true]) )
```

- Se pueden sustituir los valores null por ceros para poder calcular la diferencia también para los países que en uno de los dos años no han realizado compras
- Se crea una columna nueva con la diferencia entre los campos de suma por país

```
( = Table.AddColumn("#Valor reemplazado1", "Personalizado", each [Recuento]-[Año 2022.Recuento]) )
```

El resultado es:

	Año 2021.ventas.companies.country	1.2 Recuento	Año 2022.ventas.companies.country	1.2 Año 2022.Recuento	ABC Personalizado
1	Australia	2138,45	Australia	646,18	1492,27
2	Belgium	2308,85	Belgium	657,07	1651,78
3	Canada	14711,03	Canada	1737,49	12973,54
4	China	527,23	China	155,44	371,79
5	France	824,3	France	250,89	573,41
6	Germany	24538,06	Germany	4277,92	20260,14
7	Ireland	13433,94	Ireland	3759,18	9674,76
8	Italy	2361,01	Italy	1066,02	1294,99
9	New Zealand	1727,55	New Zealand	717,5	1010,05
10	Norway	14525,15	Norway	2761,56	11763,59
11	Sweden	18964,08	Sweden	1624,51	17339,57
12	United Kingdom	20782,4	United Kingdom	6290,77	14491,63
13	United States	4371,52	United States	884,53	3486,99
14	Netherlands	4554,32	0	0	4554,32
15	Spain	106,8	0	0	106,8

## Generación de la tabla mediante código DAX:

✕

✓

```

1 Diferencia 2021-2022 =
2 VAR Table2021 =
3     SUMMARIZECOLUMNS(
4         'ventas companies'[country],
5         "ventas2021", CALCULATE(SUM('ventas transactions'[amount]), YEAR('ventas transactions'[timestamp]) = 2021)
6     )
7
8 VAR Table2022 =
9     SUMMARIZECOLUMNS(
10        'ventas companies'[country],
11        "ventas2022", CALCULATE(SUM('ventas transactions'[amount]), YEAR('ventas transactions'[timestamp]) = 2022)
12    )
13
14 RETURN
15     ADDCOLUMNS(
16         NATURALINNERJOIN(Table2021, Table2022),
17         "Diferencia", [ventas2021] - [ventas2022]
18     )

```

country	Diferencia	ventas2021	ventas2022
Germany	20260,14	24538,06	4277,92
Australia	1492,27	2138,45	646,18
United States	3486,99	4371,52	884,53
New Zealand	1010,05	1727,55	717,5
Norway	11763,59	14525,15	2761,56
United Kingdom	14491,63	20782,4	6290,77
Italy	1294,99	2361,01	1066,02
Belgium	1651,78	2308,85	657,07
Sweden	17339,57	18964,08	1624,51
Ireland	9674,76	13433,94	3759,18
China	371,79	527,23	155,44
Canada	12973,54	14711,03	1737,49
France	573,41	824,3	250,89

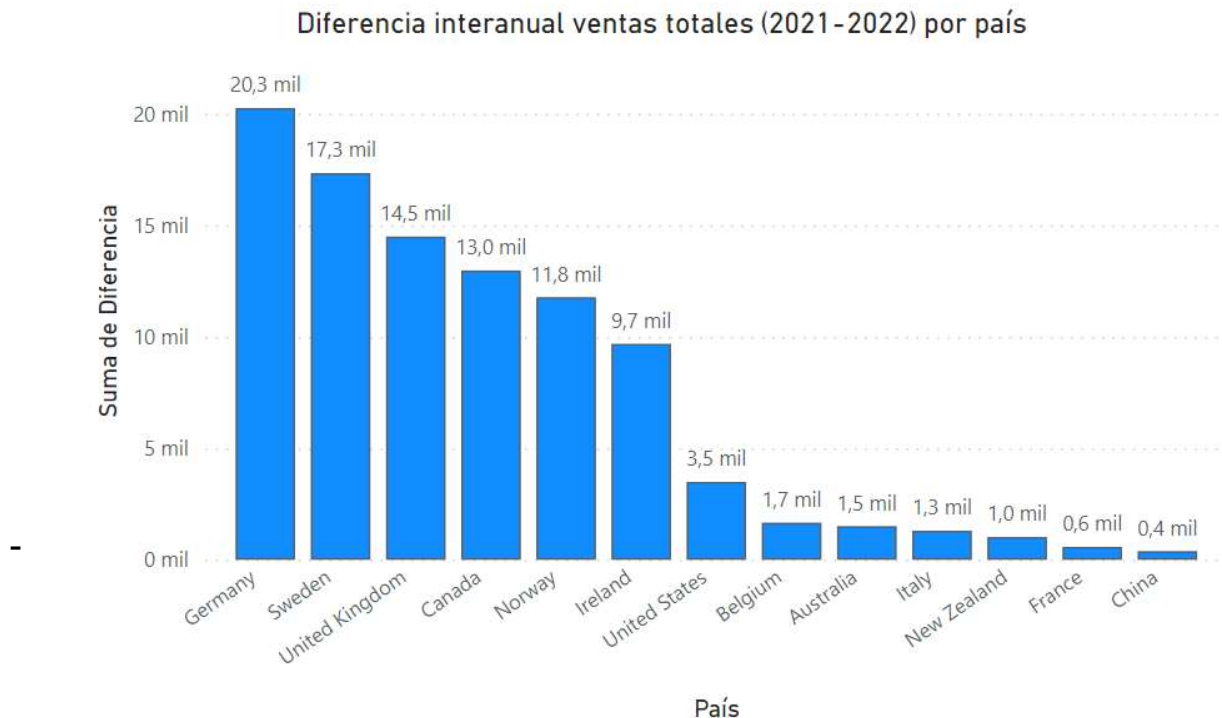
## Generación de la tabla mediante código SQL:

```
CREATE VIEW `Diferencia 2021-2022` as
SELECT y1.País1 AS País, (valor2-valor1) as Diferencia
FROM
    (SELECT country AS País1, SUM(amount) AS valor1
    FROM transactions
    JOIN companies ON company_id = business_id
    WHERE YEAR(timestamp) = 2022
    GROUP BY country) AS y1
JOIN
    (SELECT country AS País2, SUM(amount) AS valor2
    FROM transactions
    JOIN companies ON company_id = business_id
    WHERE YEAR(timestamp) = 2021
    GROUP BY country) AS y2
ON y1.País1 = y2.País2
```

ORDER BY País

;

Finalmente, se representa la diferencia de las cantidades correspondientes a los dos años por país en un gráfico de barras:

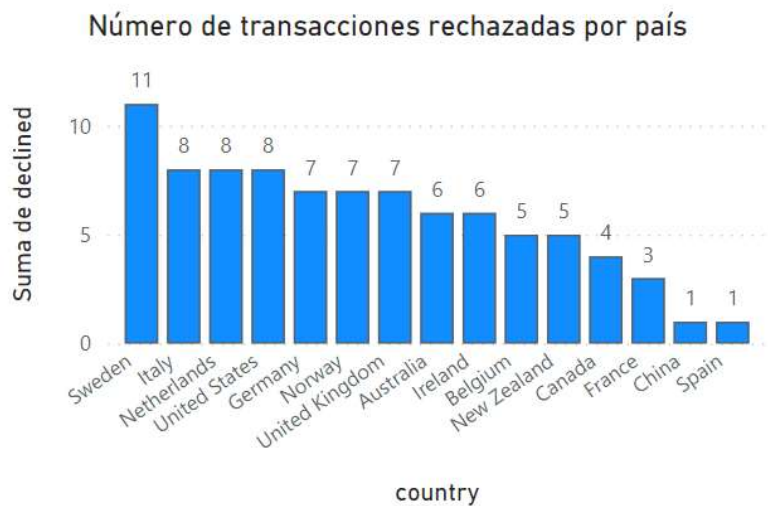


#### Ejercicio 4

Crea una visualización en la que se pueda contabilizar el número de transacciones desechadas en cada país para medir la eficacia de las operaciones. Recuerda que la empresa espera tener menos de 5 transacciones rechazadas por país.

#### Solución

Se puede hacer un gráfico de barras, directamente con la suma de declined por país sin necesidad de filtrar por valores de declined = 1 ya que al parecer, el programa lo hace automáticamente (se comprueba mediante código sql que los valores realmente corresponden a los rechazos).



```

22 • select country as País, sum(declined) as Rechazadas
23 from companies
24 join transactions on business_id = company_id
25 where declined = 1
26 group by country
27 order by Rechazadas desc;
28

```

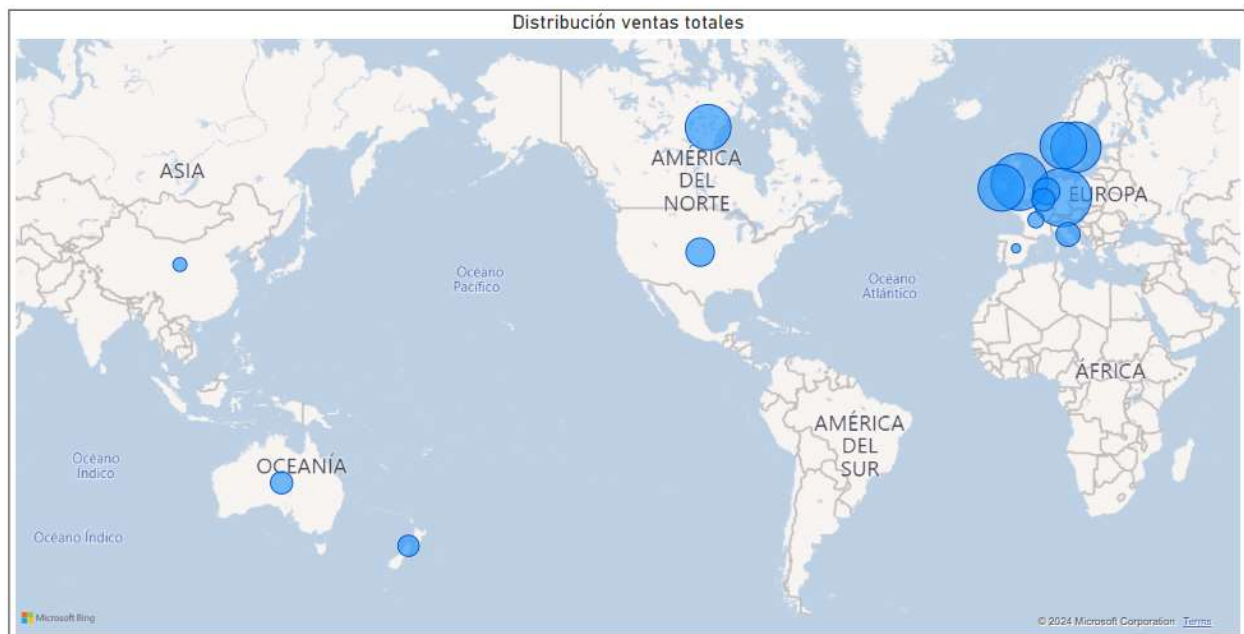
País	Rechazadas
Sweden	11
United States	8
Netherlands	8
Italy	8
Germany	7
Norway	7
United Kingdom	7
Australia	6
Ireland	6
New Zealand	5
Belgium	5
Canada	4
France	3
Spain	1
China	1

## - Ejercicio 5

La empresa busca comprender la distribución geográfica de las ventas para identificar patrones y oportunidades específicas en cada región. Selecciona la mejor vista para mostrar esta información.

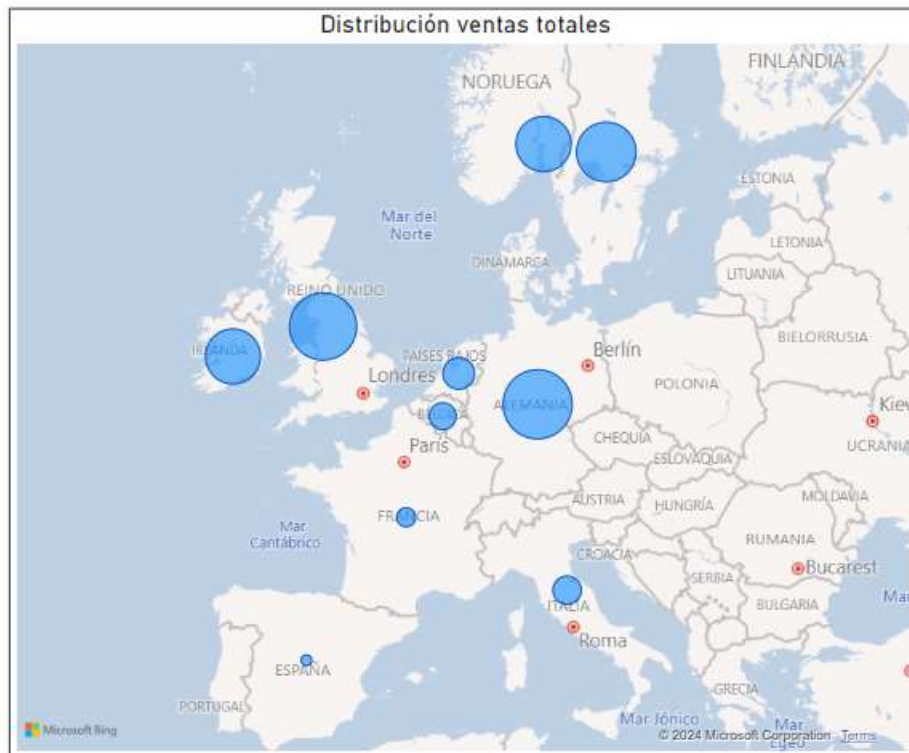
## Solución

Una buena solución sería un mapa con burbujas de tamaño proporcional a las ventas:



Claramente las ventas mayores se concentran en Europa. Para ver el detalle, se hace zoom de esta zona:





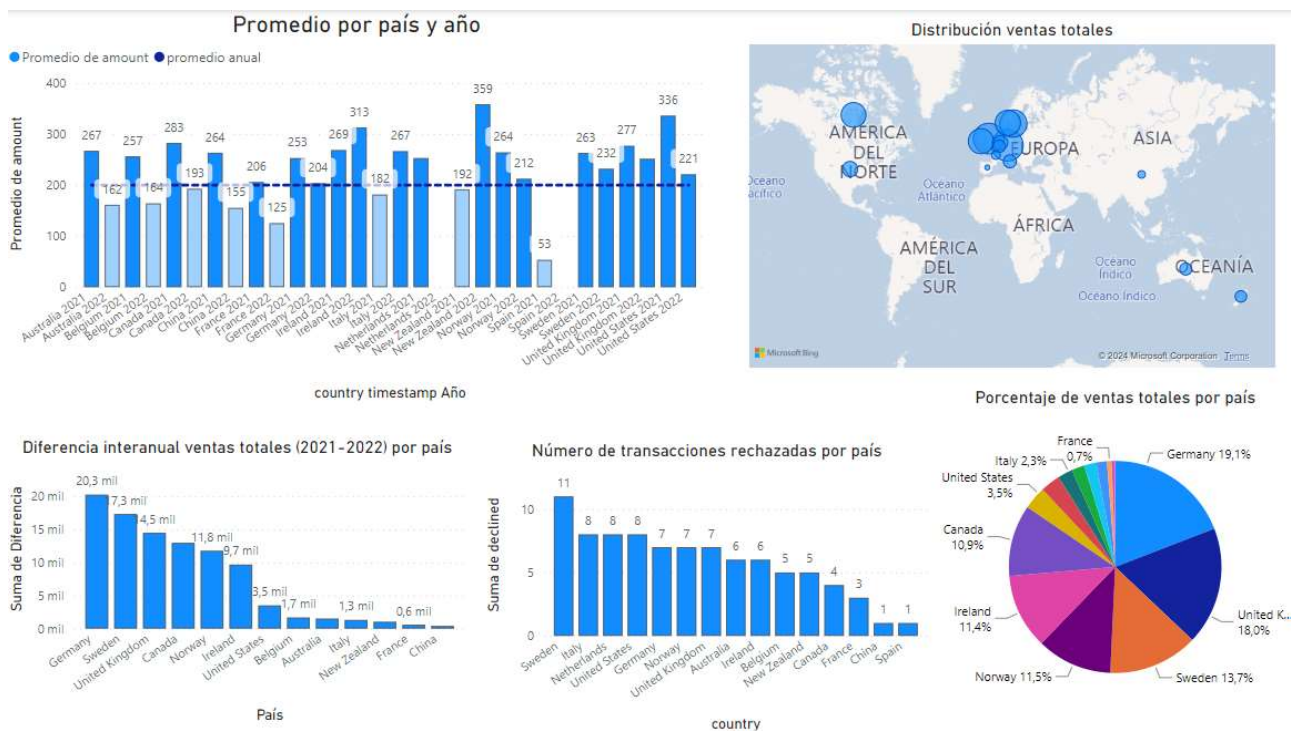
## - Ejercicio 6

Tu jefe te ha pedido preparar una presentación para tu equipo en la que se detalle la información de todos los gráficos visualizados hasta ahora. Para cumplir con esta solicitud, debes proporcionar una interpretación de las visualizaciones obtenidas. La presentación puede realizarse con información general o seleccionando un elemento en particular, como por ejemplo, los resultados de España.

## Solución

Los siguientes gráficos se han introducido en una presentación (archivo Tarea S6.01-presentación-Radostin Pavlov.pdf):





## Nivel 2

### Ejercicio 1

Su tarea consiste en implementar un filtro interactivo que permita seleccionar las ventas para cada año.

### Ejercicio 2

La gerencia está interesada en analizar más a fondo las ventas en relación al mes. Por tanto, te piden que hagas los ajustes necesarios para mostrar la información de esta manera.

### Ejercicio 3

Visualiza el total de ventas y la cantidad de transacciones realizadas. Si es necesario, puedes crear dos vistas separadas.

### Ejercicio 4

Crea una visualización que permita observar de forma efectiva y clara la cantidad de las ventas realizadas y la cantidad de transacciones rechazadas.

### Ejercicio 5

Selecciona una visualización en la que se refleje los estadísticos descriptivos de las empresas que realizaron transacciones. Recuerda mostrar el total de cada estadística.

## Solución combinada para los 5 ejercicios de este nivel

Los valores de venta y de número de transacciones se representan en un objeto de tipo tarjeta. Los filtros se han implementado en objetos de tipo matriz, ya que esta permite cambiar la ordenación de filas en columnas y vice versa. Para el filtro de “Rechazadas”, para evitar el uso de ceros y unos, se usan valores calculados con:

Rechazado = IF('ventas transactions'[declined] = 0, "No", "Sí")

El funcionamiento de los filtros se ilustra con la siguiente selección múltiple, representando las ventas y el número de transacciones no rechazadas correspondientes a los meses mayo, junio, julio y septiembre del año 2021.

Año (Ej. 1)
 

2021 2022

enero febrero marzo abril mayo junio julio agosto septiembre octubre noviembre diciembre

(Ej. 2)

Valor total ventas (Ej. 1,2,3)  

## 41,93 mil

Número de transacciones (Ej. 3)  

## 157

Rechazadas
 

No Sí

(Ej. 4)

## Nivel 3

### Ejercicio 1

En tu empresa, desean profundizar en el análisis de las características de los usuarios que participan en las transacciones, así como en los productos vendidos. Te han pedido que crees visualizaciones relevantes para mejorar estratégicamente las campañas publicitarias y aumentar las ventas. Las visualizaciones que debes incluir son las siguientes:

Información personal de los usuarios/as.

- Cantidad de transacciones realizadas y rechazadas. La empresa espera que cada usuario tenga al menos 10 transacciones por año, y que tengan menos de 2 transacciones rechazadas por año.
- Identificación del producto más barato y más caro comprado por cada usuario, junto con su precio.
- Distribución geográfica de los usuarios/as.
- Media de ventas realizada.
- El usuario/a debe tener la opción de seleccionar si desea mirar la información de un año únicamente.

Además, en este ejercicio te piden que propongamos al menos una visualización que no esté por defecto en Power BI y que amplíe la información sobre las transacciones por usuario/a. Después de crear los gráficos, debes presentar la información del usuario con el ID 96 con una breve descripción de los datos a través de una presentación de diapositivas. Asegúrate de optimizar la legibilidad y comprensión de las visualizaciones mediante ajustes adecuados.

## Solución

Para el primer punto conviene crear una tabla con el número de usuario, recuento de transacciones por usuario, recuento de transacciones rechazadas por cada usuario y año. De nuevo, esto se puede hacer mediante Power Query, DAX, y SQL.

### Código SQL:

```
38 • CREATE VIEW `Trans totales y rechazadas` AS
39 SELECT
40     users.id AS `ID de usuario`,
41     COUNT(transactions.id) AS `Total transacciones`,
42     SUM(CASE WHEN transactions.declined = 1 THEN 1 ELSE 0 END) AS `Transacciones rechazadas`,
43     YEAR(transactions.timestamp) AS `Año`
44 FROM
45     users
46 JOIN
47     transactions ON users.id = transactions.user_id
48 GROUP BY
49     users.id, YEAR(transactions.timestamp)
50 ORDER BY
51     `ID de usuario`;
```

### Código DAX:

Se crea un campo 'Year' calculado como:

Año = YEAR('ventas transactions'[timestamp])

y después se crea una nueva tabla en vista modelo con el siguiente código:

X ✓		<pre> 1 Trans totales y rechazadas = 2 SUMMARIZECOLUMNS( 3     'ventas users'[id], 4     'ventas transactions'[Año], 5     "Total transacciones", COUNT('ventas transactions'[id]), 6     "Transacciones rechazadas", CALCULATE(SUMX('ventas transactions', IF('ventas transactions'[declined] = 1, 1, 0))) 7 ) </pre>		
id	Año	Total transacciones	Transacciones rechazadas	
170	2021	1	0	
221	2021	1	0	
172	2021	1	0	
177	2021	1	0	
173	2021	1	0	
154	2021	1	0	
163	2021	1	0	
190	2021	1	0	
149	2021	1	0	
212	2021	1	0	
153	2021	1	0	
147	2021	1	0	
175	2021	1	0	
157	2021	1	0	
213	2021	1	0	
156	2021	1	0	
178	2021	1	0	
169	2021	1	0	
184	2021	1	0	
160	2021	1	0	
146	2021	1	0	
176	2021	1	0	

### Procedimiento resumido de Power Query:

- Se seleccionan los campos timestamp, declined y user\_id de la tabla transactions
- Se crea una columna nueva con el año (Date.Year([timestamp]))
- La tabla se duplica
- Una de las tablas se filtra por año 2021 y la otra por 2022
- Las dos tablas se duplican
- Las nuevas tablas se filtran por declined = 1

Ahora tenemos dos tablas con las transacciones totales (una tabla por cada año) y dos tablas con las transacciones rechazadas (una tabla por cada año también)

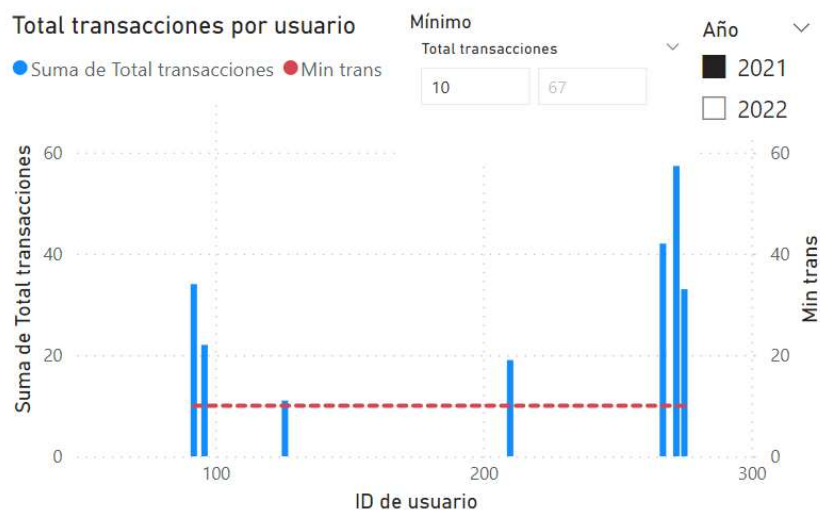
- En las 4 tablas se aplica recuento de filas agrupando por user\_id
- En cada tabla se añade una columna con el año correspondiente ya que esta fue borrada al hacer el recuento
- Se hace left join entre las tablas del mismo año (left join porque la primera tabla contiene recuento de todas las transacciones, mientras que la segunda contiene recuento solo de las ttransacciones rechazadas)
- Se añade de nuevo la columna con el año correspondiente a cada una de las dos tablas join-eadas
- Los valores null se reemplazan por ceros
- Se unifican los nombres de las columnas y las dos tablas se anexan (equivalente a UNION)

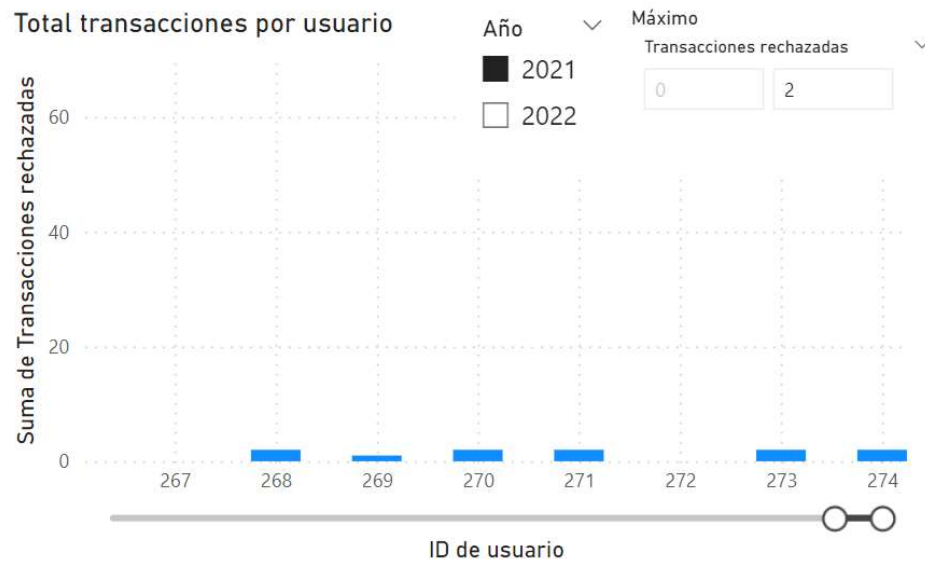
La tabla final tiene este aspecto:

	123 user_id	123 total tr	1.2 declined tr	ABC 123 Año
1	272	67	32	2021
2	267	42	12	2021
3	92	34	0	2021
4	275	33	19	2021
5	96	22	0	2021
6	210	19	0	2021
7	126	11	0	2021
8	110	5	0	2021
9	271	4	2	2021
10	268	4	2	2021
11	270	3	2	2021
12	269	3	1	2021
13	274	3	2	2021
14	273	3	2	2021
15	251	2	0	2021
16	224	2	0	2021
17	242	2	0	2021
18	232	2	0	2021
19	246	2	0	2021
20	264	2	0	2021
21	201	2	0	2021
22	211	2	0	2021
23	255	2	0	2021
24	240	2	0	2021
25	225	2	0	2021
26	204	2	0	2021
27	233	2	0	2021
28	209	2	0	2021

A continuación, a partir de la tabla generada se crean dos visualizaciones:

- Total de transacciones por usuario, con filtro interactivo por año y mínimo de transacciones deseado
- Número de transacciones rechazadas, con filtro interactivo por año y máximo de transacciones rechazadas





Para la identificación del producto más barato y más caro comprado por cada usuario, junto con su precio, conviene crear una tabla que contenga los siguientes campos: user\_id, tipo de precio (mínimo o máximo), precio, product\_id, product\_name, año. Esta tabla se ha generado de dos maneras: con Power Query y con SQL.

### Procedimiento Power Query

- Se duplica la tabla transactions y la tabla generada se denomina productos\_por\_usuario
- Se expanden los campos necesarios para acceder a los campos deseados:

```
= Table.ExpandTableColumn(ventas_transactions, "ventas.products_transactions",
{"product_id", "ventas.products"}, {"ventas.products_transactions.product_id",
"ventas.products_transactions.ventas.products"})
```

```
= Table.ExpandRecordColumn("#Se expandió ventas.products_transactions",
"ventas.products_transactions.ventas.products", {"product_name", "price"},
{"ventas.products_transactions.ventas.products.product_name",
"ventas.products_transactions.ventas.products.price"})
```

- Se eliminan las columnas innecesarias:

```
= Table.SelectColumns("#Se expandió ventas.products_transactions.ventas.products",
{"timestamp", "user_id", "ventas.products_transactions.product_id",
"ventas.products_transactions.ventas.products.product_name",
"ventas.products_transactions.ventas.products.price"})
```

- Se crea una columna, eliminando el símbolo \$ del precio y convirtiendo el resultado en número

```
= Table.AddColumn("#Columnas reordenadas", "Personalizado", each
Text.Middle([ventas.products_transactions.ventas.products.price], 1,
Text.Length([ventas.products_transactions.ventas.products.price]) - 1))
```

```
= Table.TransformColumnTypes("#Personalizada agregada", {"Personalizado", type number}))
```

- Se crea una columna con el año (esto es opcional, ya que siempre se puede extraer el año de la jerarquía que power BI genera automáticamente a partir del campo timestamp).

```
= Table.AddColumn("#Tipo cambiado", "Personalizado.1", each Date.Year([timestamp]))
```

- Se quitan las columnas timestamp y precio originales
- Se ponen nombres apropiados a las columnas restantes
- La tabla generada así, se duplica dos veces para crear una tabla con el precio mínimo y otra con el precio máximo por cada usuario
- Se utilizan las funciones de agregación min() y max() y se agrupa por user\_id

```
= Table.Group("#Columnas con nombre cambiado", {"user_id"}, {"Recuento", each List.Min([precio_USD]), type nullable number}))
```

- A cada tabla se le agrega un campo llamado tipo\_precio y se pone valor “Min” o “Max” para las dos tablas respectivas.

```
= Table.AddColumn("#Filas ordenadas", "Personalizado", each "Min")
```

- Se unifican los nombres de las dos tablas (la de precios mínimos y la de precios máximos)
- Las dos tablas se anexan (equivalente de Union) generando una nueva tabla, llamada min-max que contiene tres campos: user\_id, precio, tipo\_precio (Min o Max)
- Esta tabla se combina con la tabla original productos\_por\_usuario en base a los campos de precio y de user\_id, recuperando de esta manera la referencia y el nombre del producto, así como el año (estos se habían perdido al hacer la agregación).

La tabla resultante se muestra a continuación:



user_id	precio	tipo_precio	product_id	product_name	año
60	195,94	Max	67	Winterfell	2022
60	26,51	Min	83	duel tourney	2022
61	114,09	Max	73	Dorne bastard	2021
61	65,25	Min	97	jinn Winterfell	2021
62	167,2	Max	29	Tully maester Tarly	2021
62	167,2	Min	29	Tully maester Tarly	2021
63	26,51	Max	83	duel tourney	2022
63	26,51	Min	83	duel tourney	2022
64	167,2	Max	29	Tully maester Tarly	2021
64	63,33	Min	7	north of Casterly	2021
65	63,33	Max	7	north of Casterly	2021
65	63,33	Min	7	north of Casterly	2021
66	167,2	Max	29	Tully maester Tarly	2021
66	26,51	Min	83	duel tourney	2021
67	91,89	Max	17	skywalker ewok sith	2021
67	49,7	Min	11	Karstark Dorne	2021
68	167,2	Max	29	Tully maester Tarly	2021
68	167,2	Min	29	Tully maester Tarly	2021
69	141,01	Max	41	Lannister Barratheon Direwolf	2021
69	26,51	Min	83	duel tourney	2021
70	49,7	Max	11	Karstark Dorne	2021
70	49,7	Min	11	Karstark Dorne	2021
71	141,01	Max	41	Lannister Barratheon Direwolf	2022
71	28,01	Min	61	Winterfell Lannister	2022
72	195,94	Max	67	Winterfell	2021
72	26,66	Min	37	Direwolf Littlefinger	2021
73	139,59	Max	13	palpatine chewbacca	2021

## Código SQL:

```

CREATE VIEW `Precios min-max usuarios` AS
WITH prices AS (
  SELECT
    transactions.user_id,
    products.id AS id_producto,
    product_name AS Producto,
    CAST(SUBSTRING(price, 2, LENGTH(price) - 1) AS DECIMAL(10, 2)) AS Precio,
    YEAR(timestamp) AS Año,
    MIN(CAST(SUBSTRING(price, 2, LENGTH(price) - 1) AS DECIMAL(10, 2))) OVER
(PARTITION BY transactions.user_id) AS min_price,
    MAX(CAST(SUBSTRING(price, 2, LENGTH(price) - 1) AS DECIMAL(10, 2))) OVER
(PARTITION BY transactions.user_id) AS max_price
  FROM transactions
  JOIN products_transactions ON products_transactions.transaction_id = transactions.id
  JOIN products ON products.id = products_transactions.product_id
)
SELECT
  CASE WHEN Precio = min_price THEN 'Min'
    WHEN Precio = max_price THEN 'Max'
  END AS Tipo_precio,
  user_id AS Usuario,
  id_producto,
  Producto,
  Precio,

```

```
Año
FROM prices
WHERE Precio = min_price OR Precio = max_price;
```

A continuación, se usa la tabla generada para crear una visualización de los datos con filtros interactivos para Año, tipo Min/Max y número de usuario:

Año

☒ 2021

☐ 2022

Tipo\_precio

☒ Max

☐ Min

Usuario

☐ 95

☒ 96

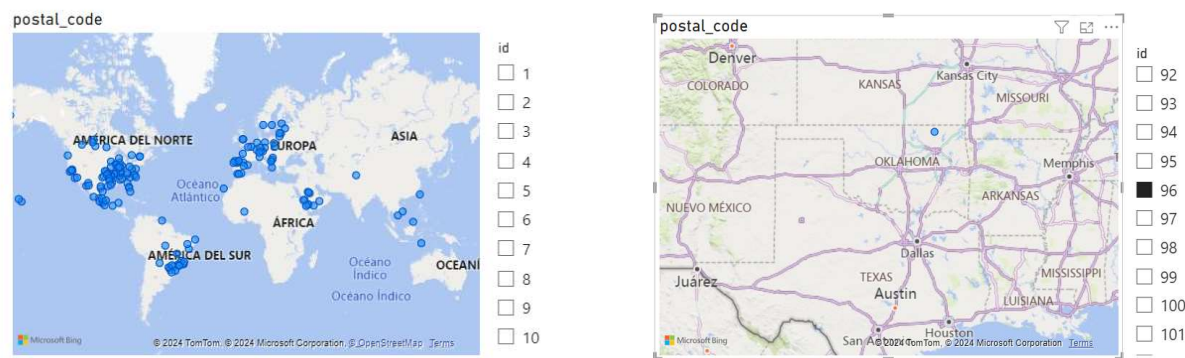
☐ 97

☐ 98

☐ 99

☐ 100

La distribución geográfica de los usuarios se ha hecho usando sus códigos postales como input para un mapa. Aquí también se ha añadido filtro interactivo para poder escoger el usuario:



postal\_code

id

☐ 92

☐ 93

☐ 94

☐ 95

☒ 96

☐ 97

☐ 98

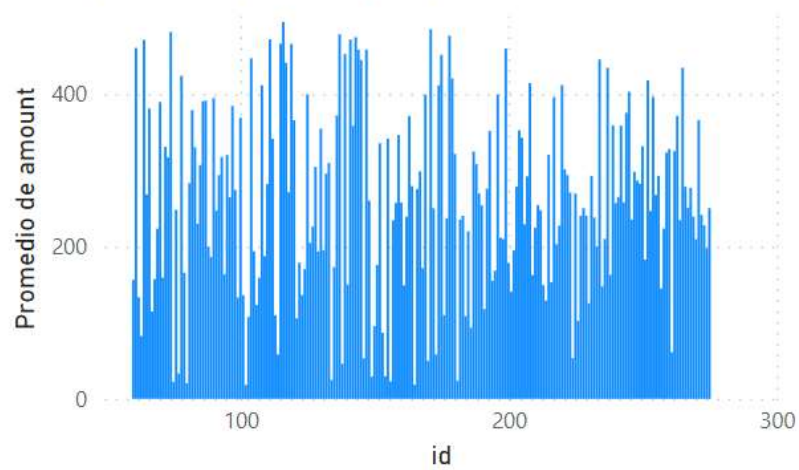
☐ 99

☐ 100

☐ 101

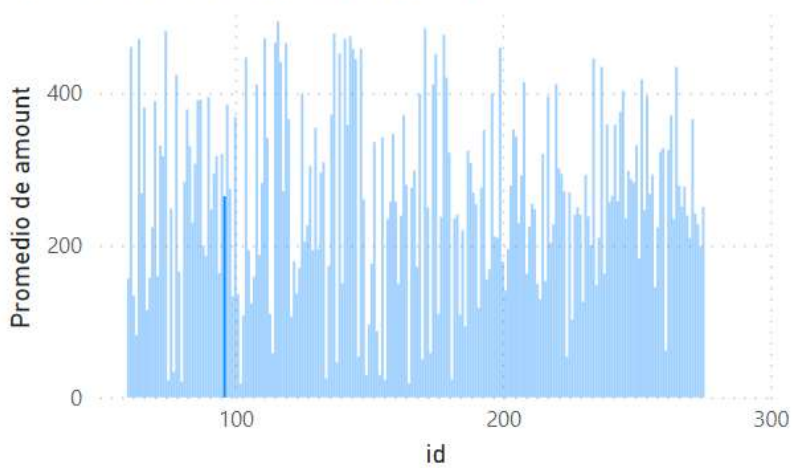
Finalmente, se ha creado un gráfico de barras y una tabla con el promedio de ventas por usuario:

Valor promedio de ventas por usuario



id	Promedio de amount
60	156,23
61	460,82
62	133,39
63	82,43
64	471,47
65	268,48
66	381,17
67	114,58
68	157,20
69	223,53
70	389,70
<b>Total</b>	<b>256,74</b>

Valor promedio de ventas por usuario



id	Promedio de amount
89	186,12
90	394,59
91	247,13
92	293,63
93	317,41
94	163,40
95	320,29
96	264,63
97	384,42
98	274,28
99	132,80
<b>Total</b>	<b>256,74</b>