

# BIG DATA





# PRESENTACIONES

- Conocimientos previos Big Data
- Conocimientos SQL
- Conocimientos MongoDB
- Conocimientos Power BI y Pentaho

# La creación de datos, a punto de explotar

Cantidad real y prevista de datos generados en todo el mundo (en zettabytes)



@Statista\_ES

Fuente: Statista Digital Economy Compass 2019

statista

# THE INTERNET IN **2023** EVERY MINUTE



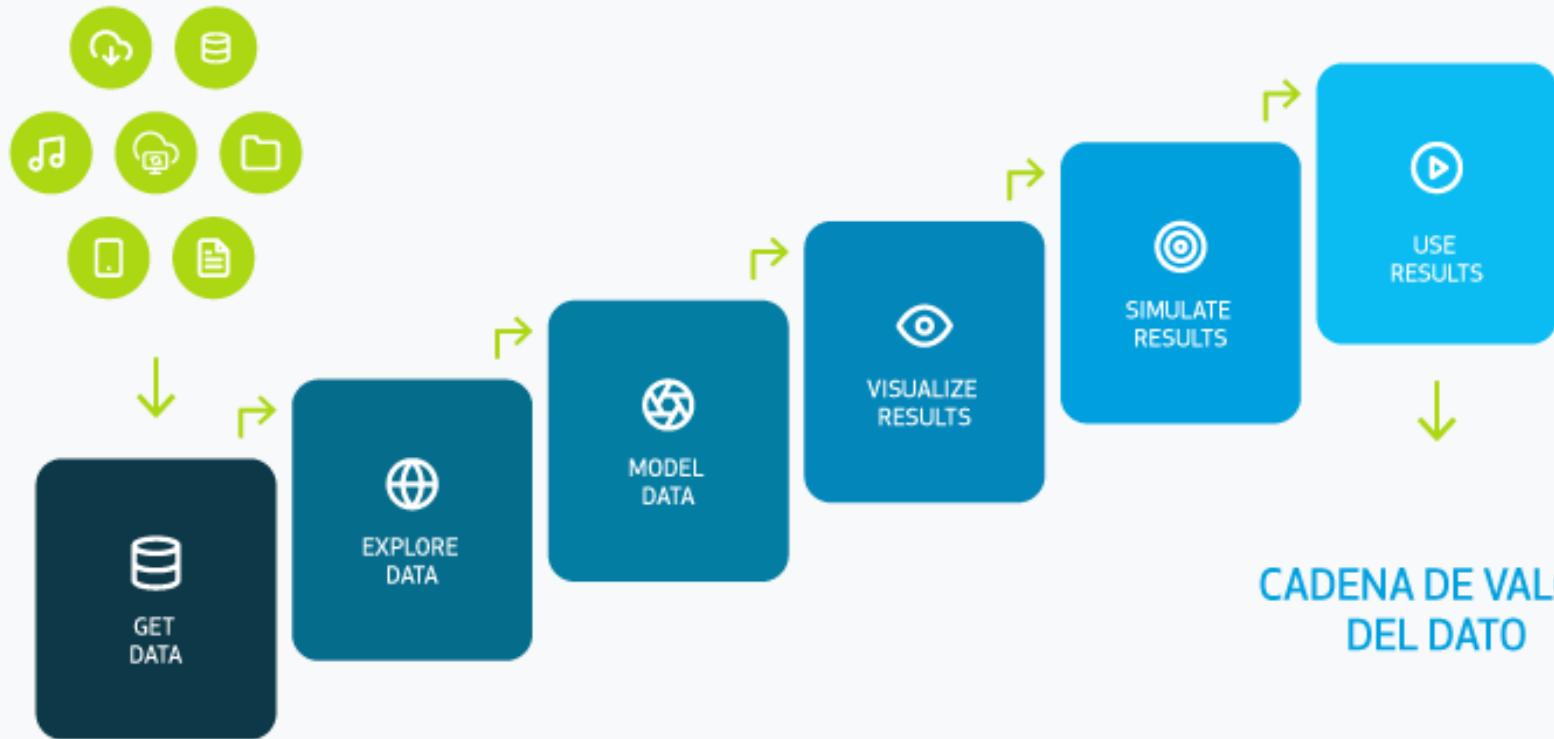
Created by: eDiscovery Today & LTMG

## Where does big data come from?

Most big data efforts are currently focused on analyzing internal data to extract insights. Fewer organizations are looking at data outside their firewalls, such as social media.

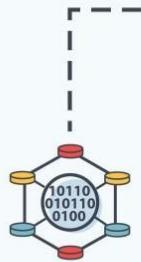


IBM





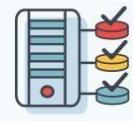
# BIG DATA



Volume



Value



Veracity



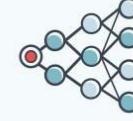
Visualization



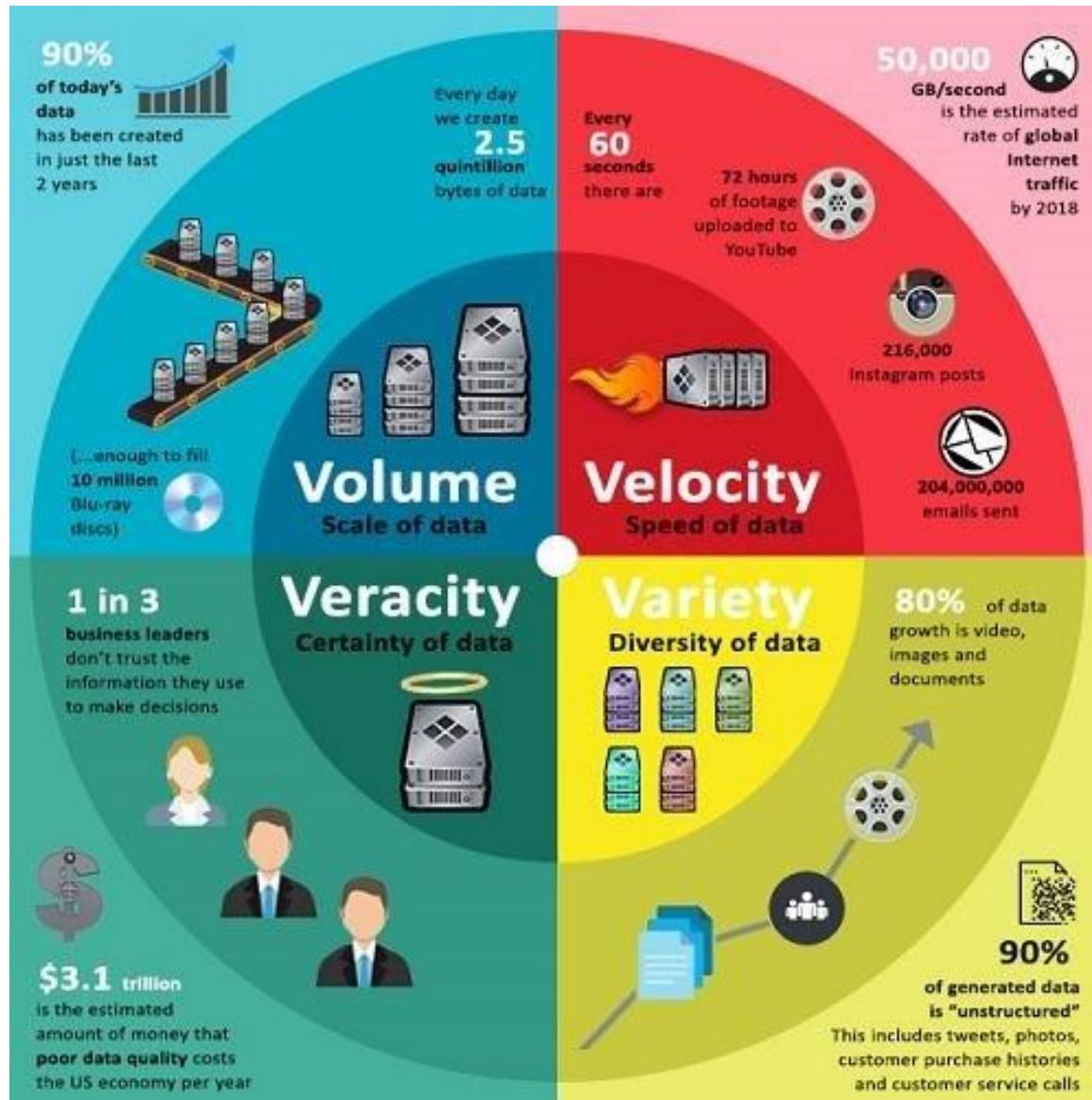
Variety

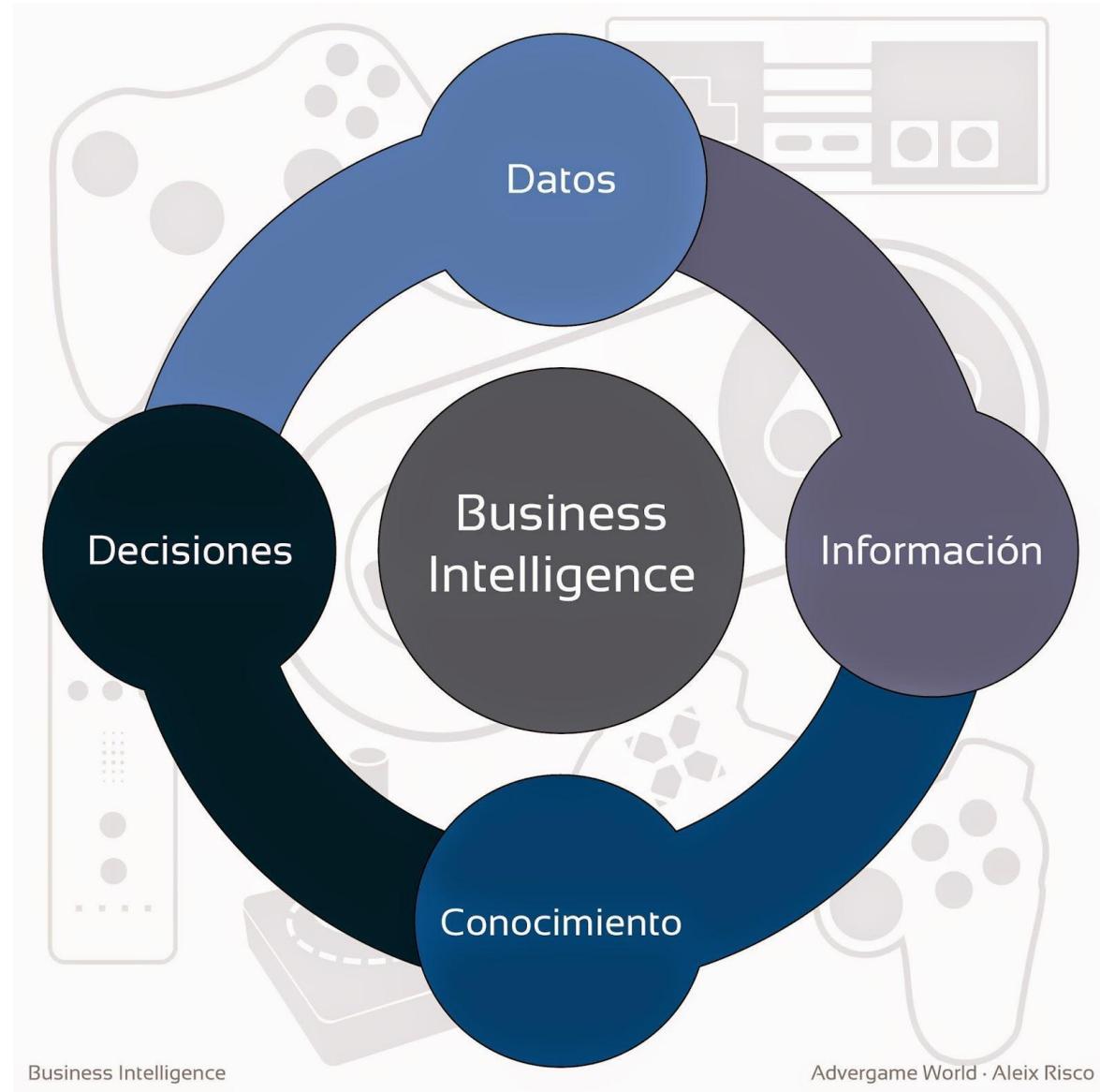


Velocity



Virality





## ¿DE DÓNDE VIENEN LOS DATOS?



GENERADOS  
POR PERSONAS



TRANSACCIONES  
DE DATOS



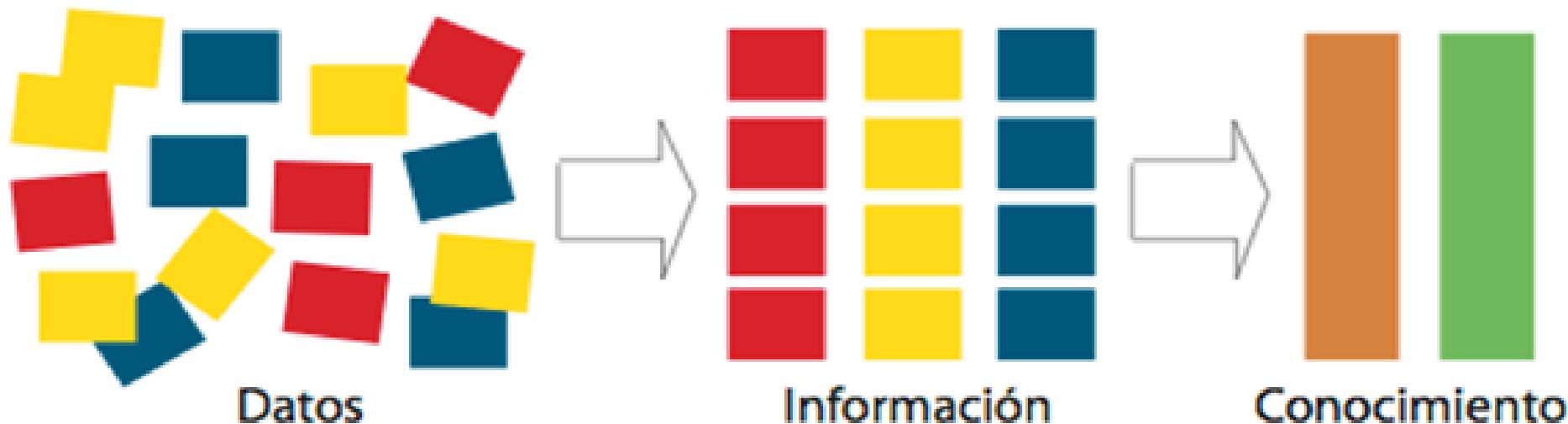
INTERNET OF  
THINGS



BIOMÉTRICA

ORIGEN DE DATOS

## Tipos de contenidos y su evolución



Uber

Coca-Cola

NETFLIX

## THE COMPANIES THAT USE ANALYTICS BEST ARE...

1 0 0 1 0 0  
1 0 0 1 1 0 0  
1 1 0 1 1 0 0  
1 1 0 1 1 0 0  
1 1 0 1 1 0 0  
1 1 0 1 1 0 0  
1 1 0 1 1 0 0  
1 1 0 1 1 0 0

**2X**

more likely  
to have  
top-quartile  
financial  
performance

1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0  
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0  
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0  
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0  
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0  
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0  
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0  
1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0

**5X**

more likely  
to make  
decisions  
“much faster”  
than  
competition

0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0

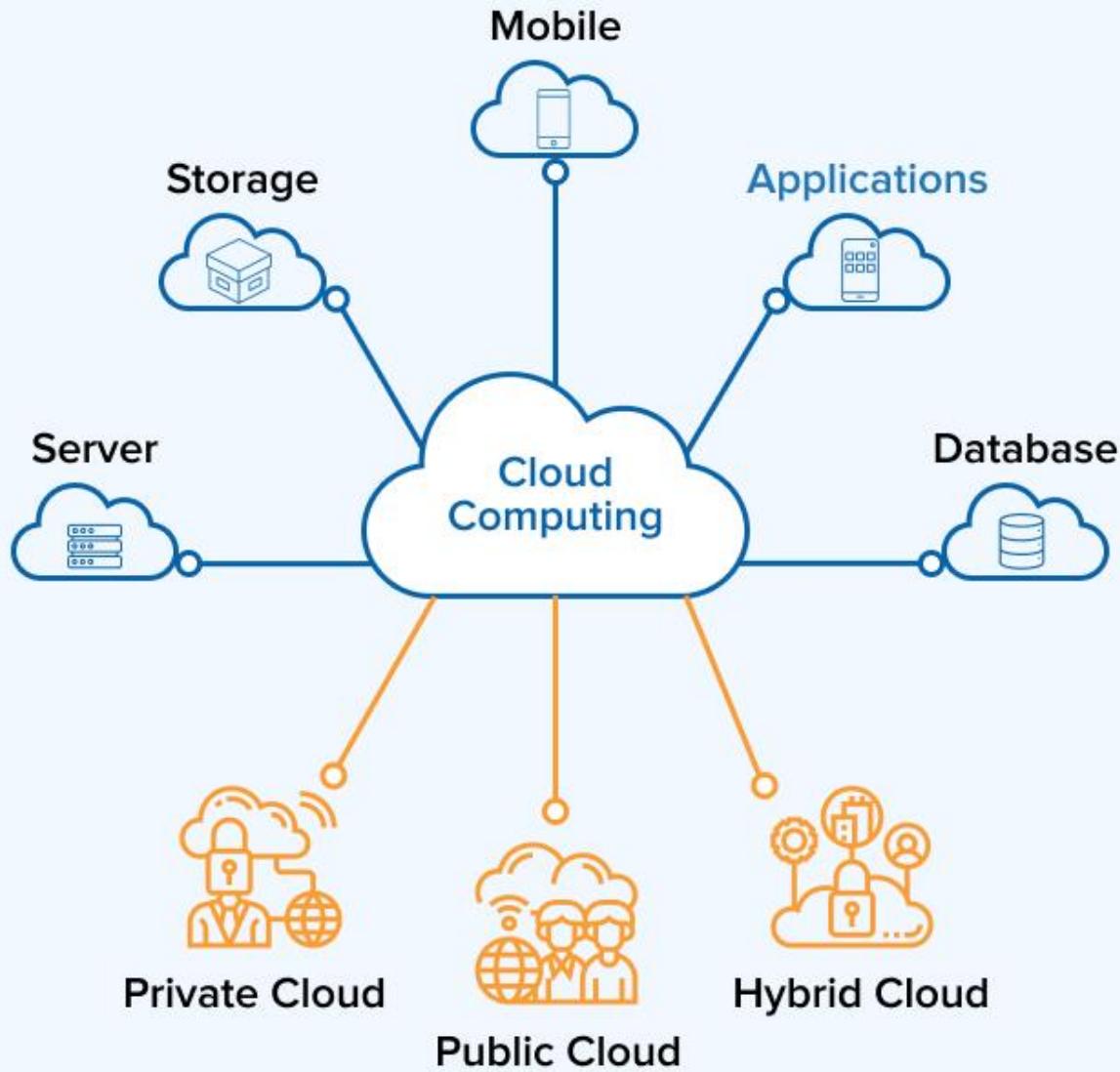
**3X**

more likely  
to execute  
decisions  
as intended

1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1  
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1

**2X**

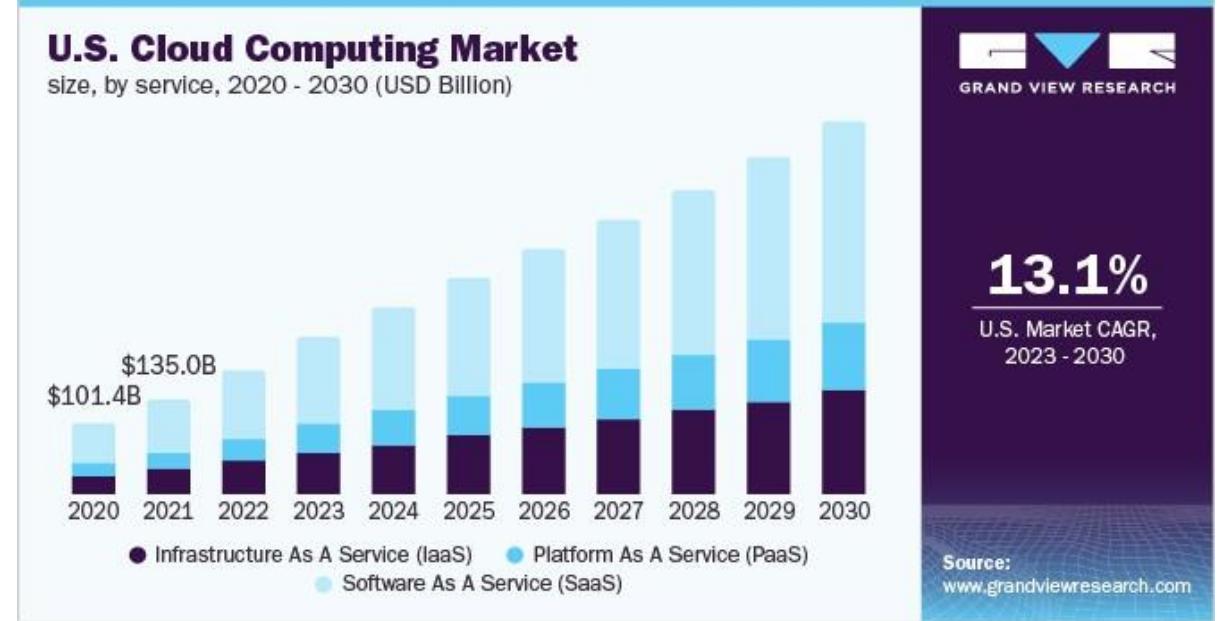
more likely to  
use data very  
frequently  
when making  
decisions



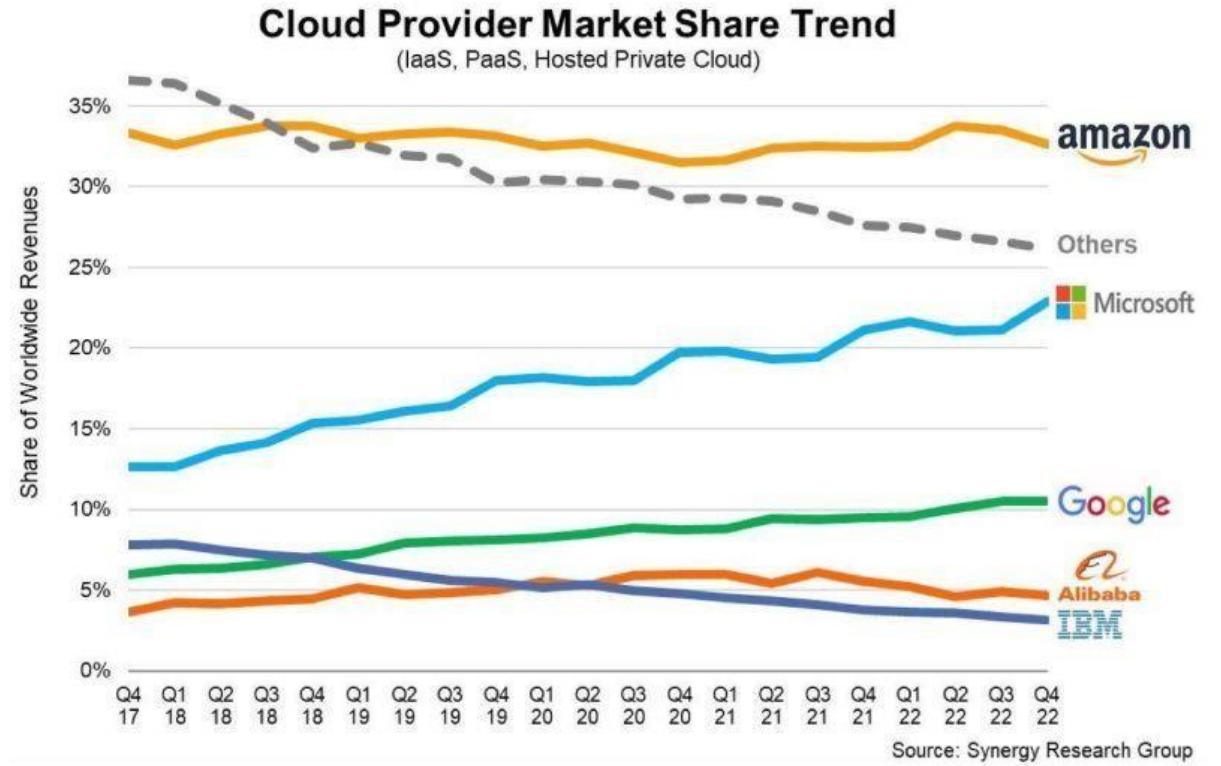
# VENTAJAS DE CLOUD COMPUTING

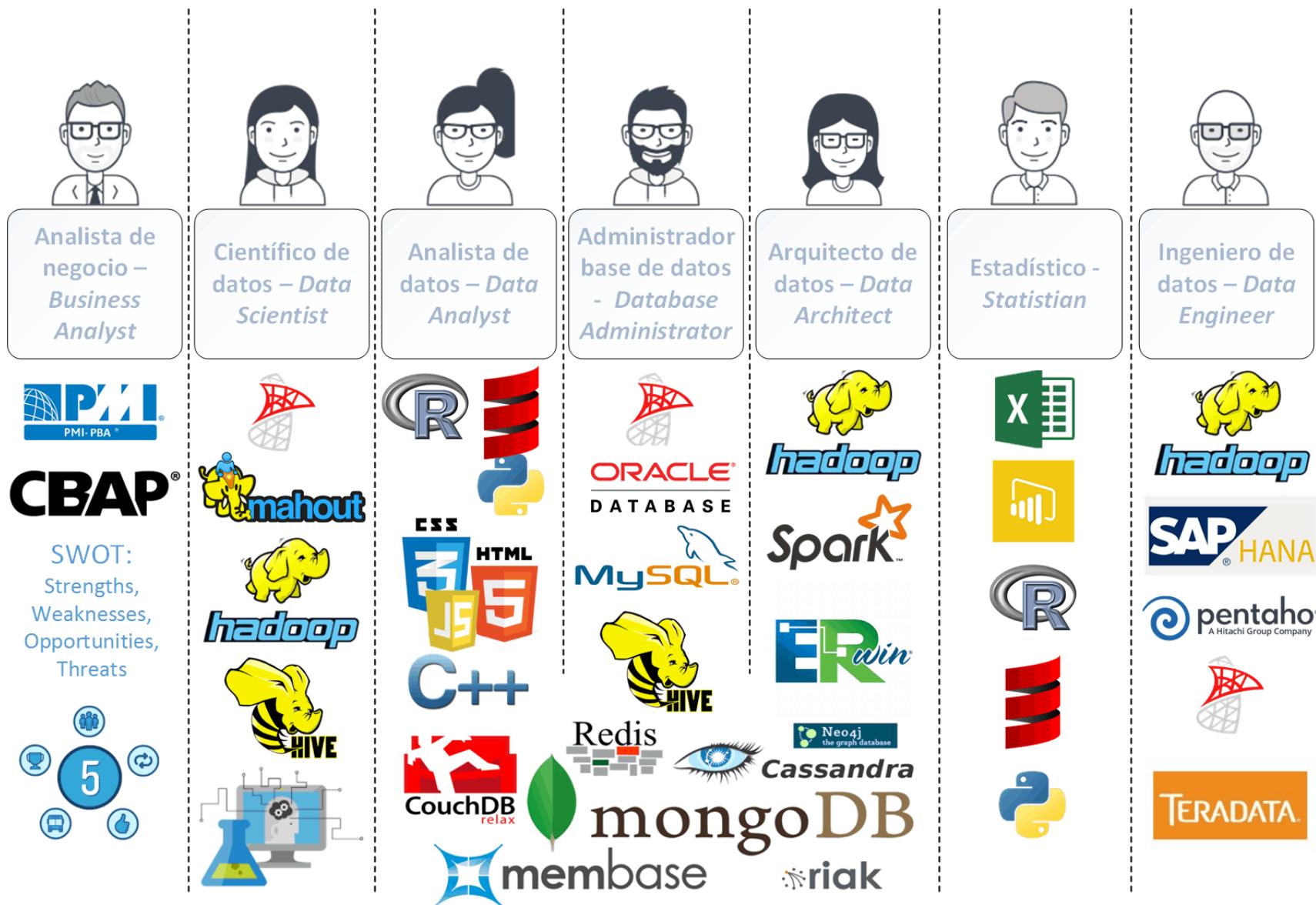


# EVOLUCIÓN CLOUD COMPUTING



# PRINCIPALES PROVEEDORES







**MICROSOFT  
FABRIC**

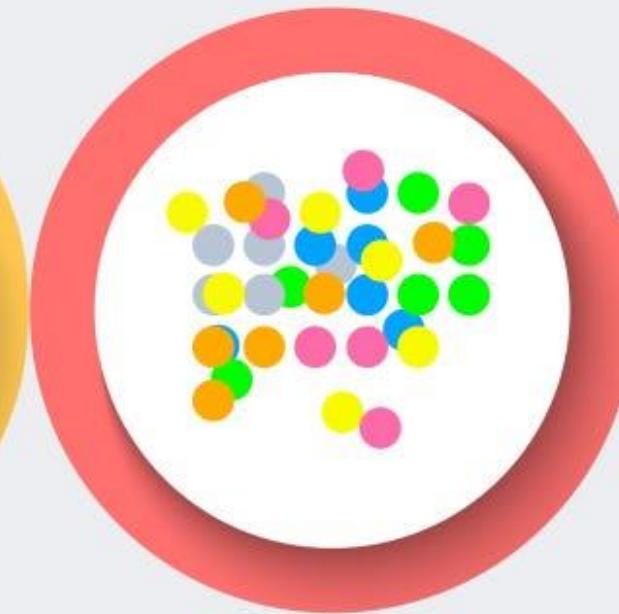
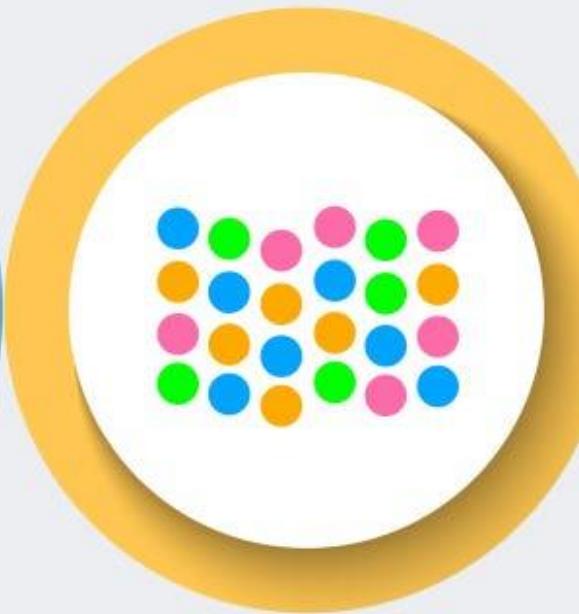
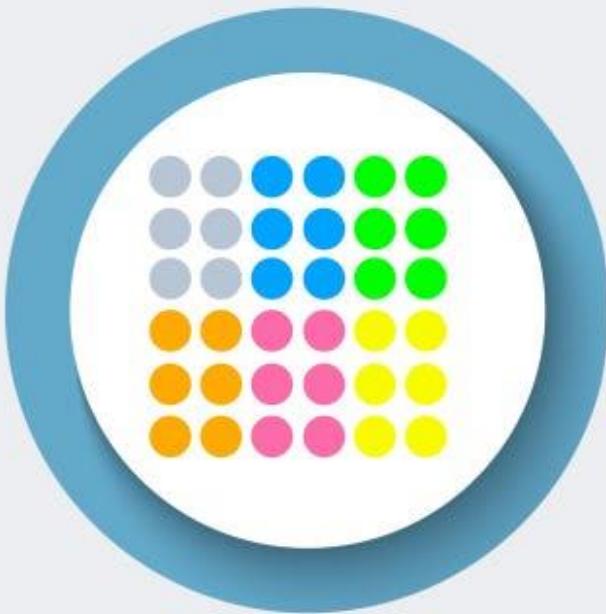
# DATOS

.....

Estructurados

Semiestructurados

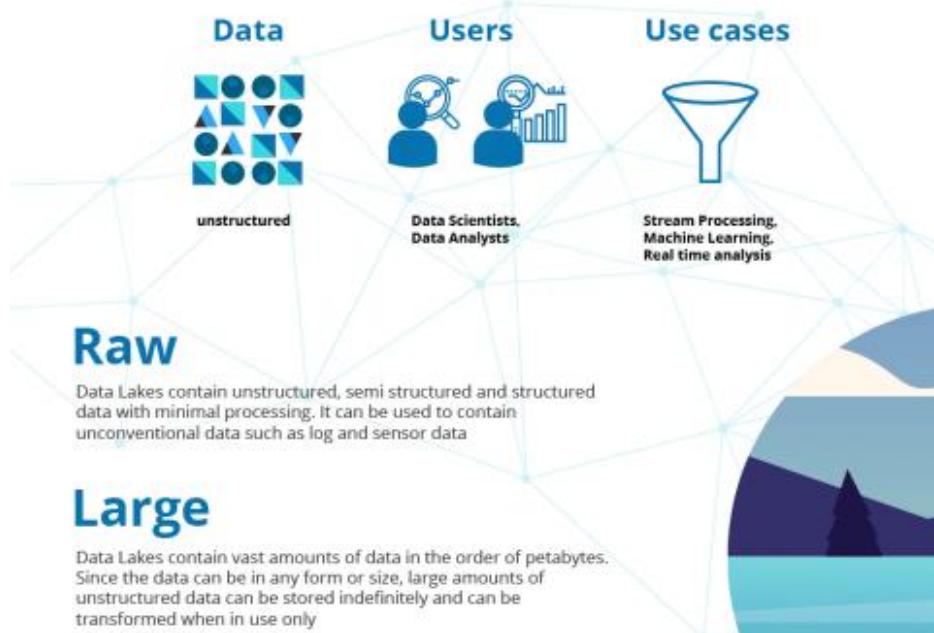
No  
Estructurados



# DATA LAKE

# vs

# DATA WAREHOUSE



## Raw

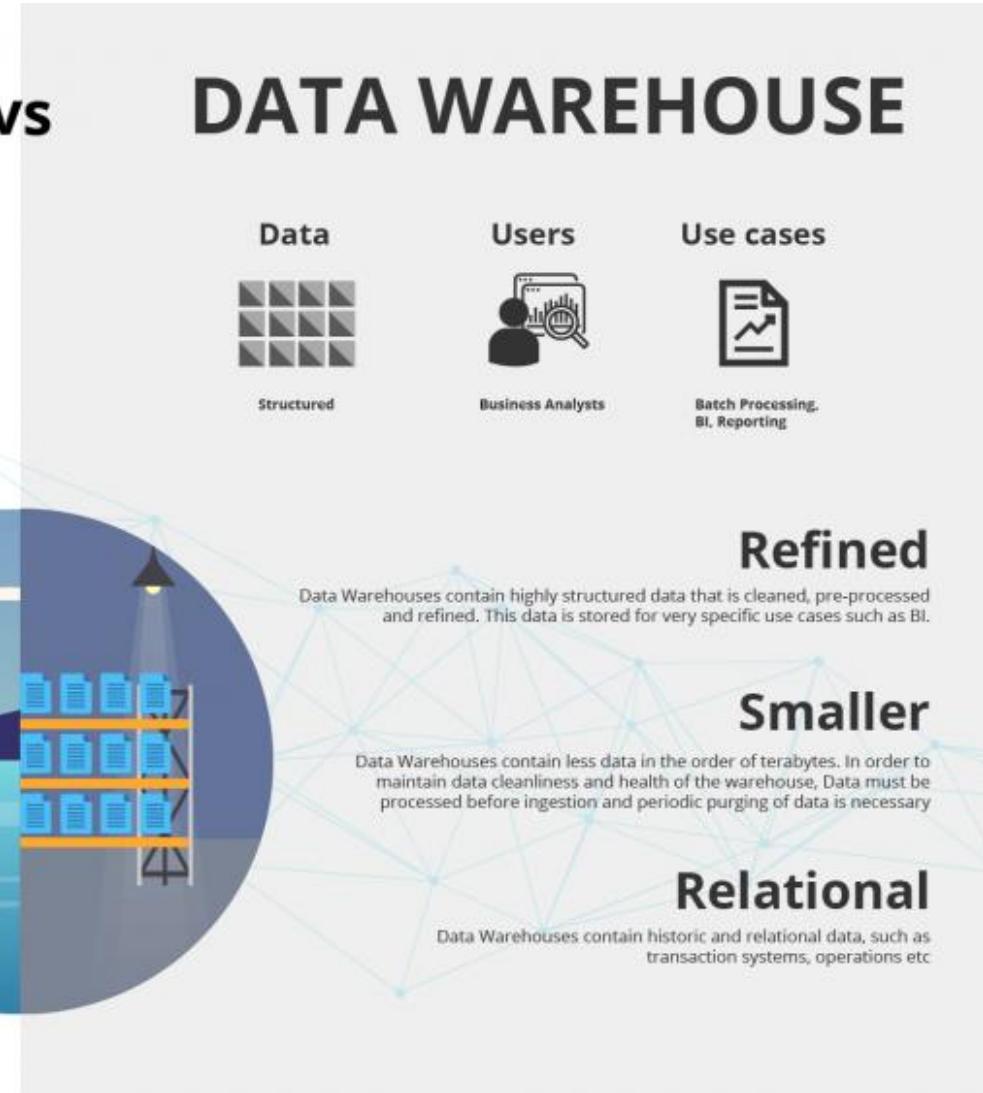
Data Lakes contain unstructured, semi structured and structured data with minimal processing. It can be used to contain unconventional data such as log and sensor data

## Large

Data Lakes contain vast amounts of data in the order of petabytes. Since the data can be in any form or size, large amounts of unstructured data can be stored indefinitely and can be transformed when in use only

## Undefined

Data in data lakes can be used for a wide variety of applications, such as Machine Learning, Streaming analytics, and AI



## Refined

Data Warehouses contain highly structured data that is cleaned, pre-processed and refined. This data is stored for very specific use cases such as BI.

## Smaller

Data Warehouses contain less data in the order of terabytes. In order to maintain data cleanliness and health of the warehouse, Data must be processed before ingestion and periodic purging of data is necessary

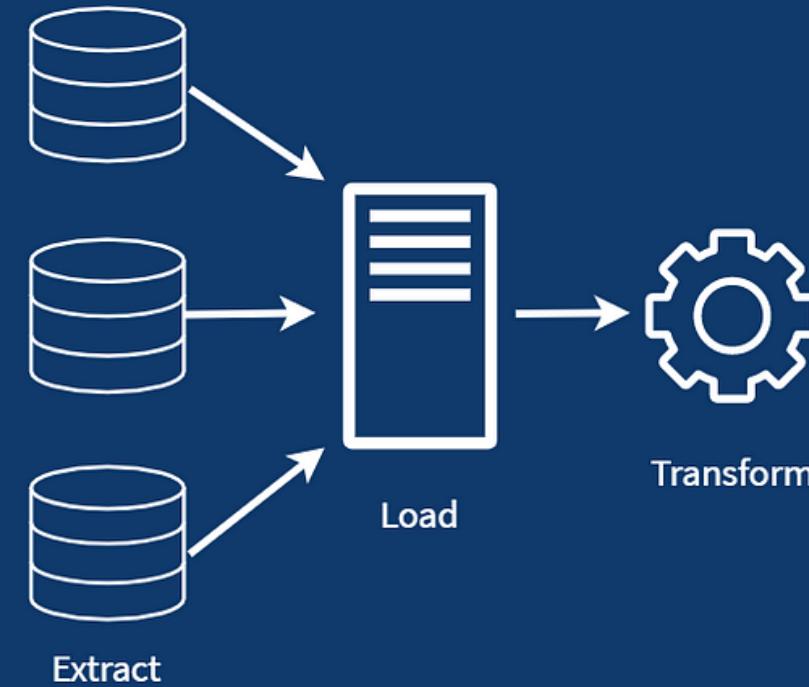
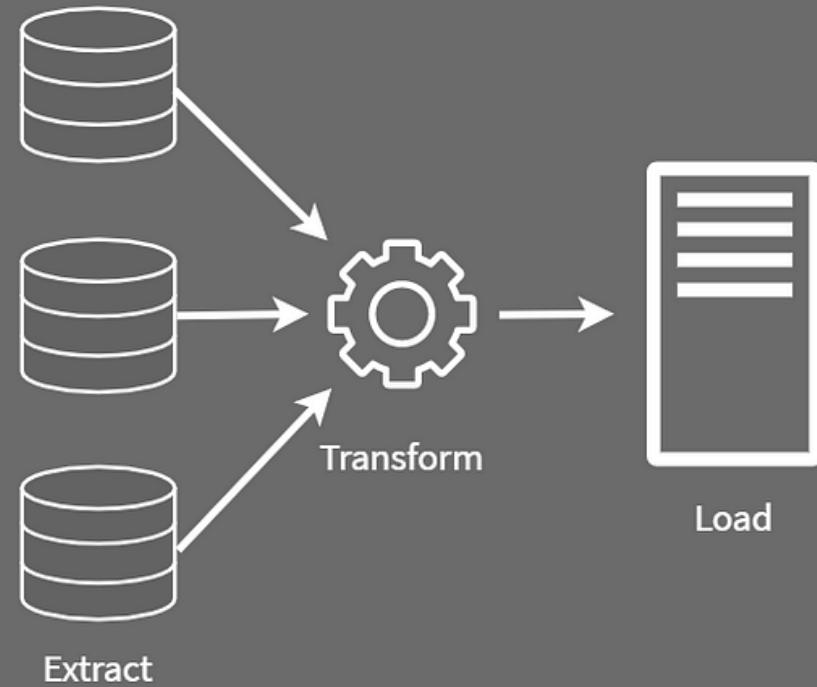
## Relational

Data Warehouses contain historic and relational data, such as transaction systems, operations etc

ETL

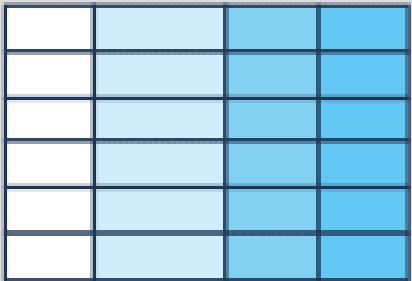
-VS-

ELT

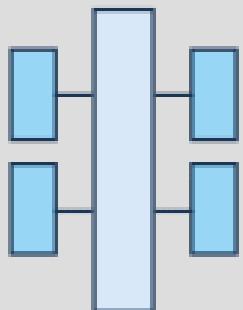


# SQL

## Relational

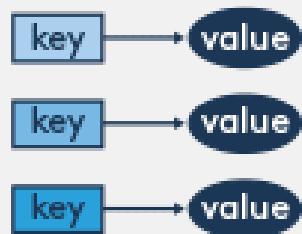


## Analytical (OLAP)

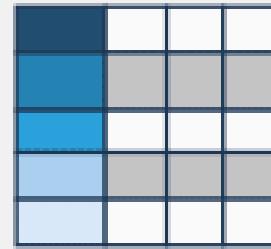


# NoSQL

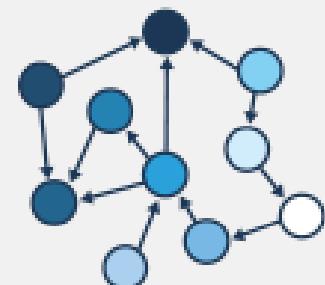
## Key-Value



## Column-Family



## Graph



## Document





The screenshot shows the Apache Atlas web interface. At the top, there is a navigation bar with the organization name "MyOrg" and a dropdown menu. Below the navigation bar, there are links for "Access Manager", "Support", and "Billing". On the right side of the top bar, there is a "All Clusters" button with a dropdown arrow. In the center of the header, there is a search bar and a cluster selection dropdown.

The main content area has a breadcrumb navigation path: "myProDev > Charts > Clusters". There are three tabs at the top of this section: "Clusters" (selected), "Triggers", and "Data Lake". A green button labeled "Create a New Cluster" is located on the right side of the header.

The main content area is titled "Clusters" and contains a search bar with the placeholder "Find a cluster...". Below the search bar, there is a card for the "MainDBCluster". The card includes the following details:

- SANDBOX**
- MainDBCluster** (Status: Green)
- Version 4.2.11
- CONNECT**, **METRICS**, **COLLECTIONS**, **...** buttons
- CLUSTER TIER**: M0 Sandbox (General)
- REGION**: AWS / N. Virginia (us-east-1)
- TYPE**: Replica Set - 3 nodes
- LINKED REALM APP**: None Linked

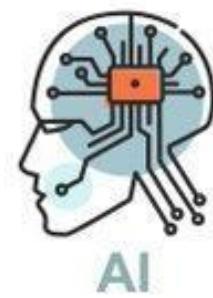
A context menu is open over the "..." button, listing options: "Edit Configuration", "Command Line Tools", "Load Sample Dataset", and "Terminate". The "Terminate" option is highlighted with a cursor icon.

On the right side of the cluster card, there are two metrics charts:

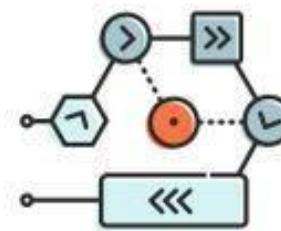
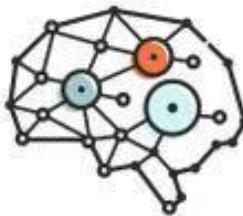
- Operations R: 0 W: 0** (100.0/s) - Last 6 Hours
- Logical Size 0.0 B** (512.0 MB max) - Last 6 Hours

A callout box on the right side of the card says "Enhance Your Experience" with the text: "For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!". A green "Upgrade" button is located at the bottom right of this callout.





DEEP LEARNING



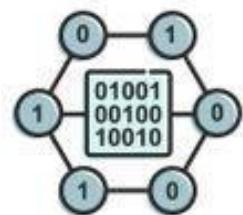
ALGORITHM



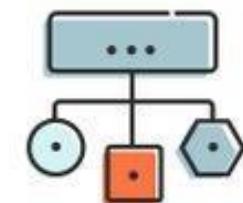
LEARNING



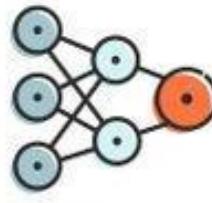
IMPROVES



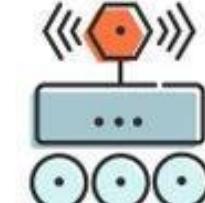
DATA MINING



CLASSIFICATION



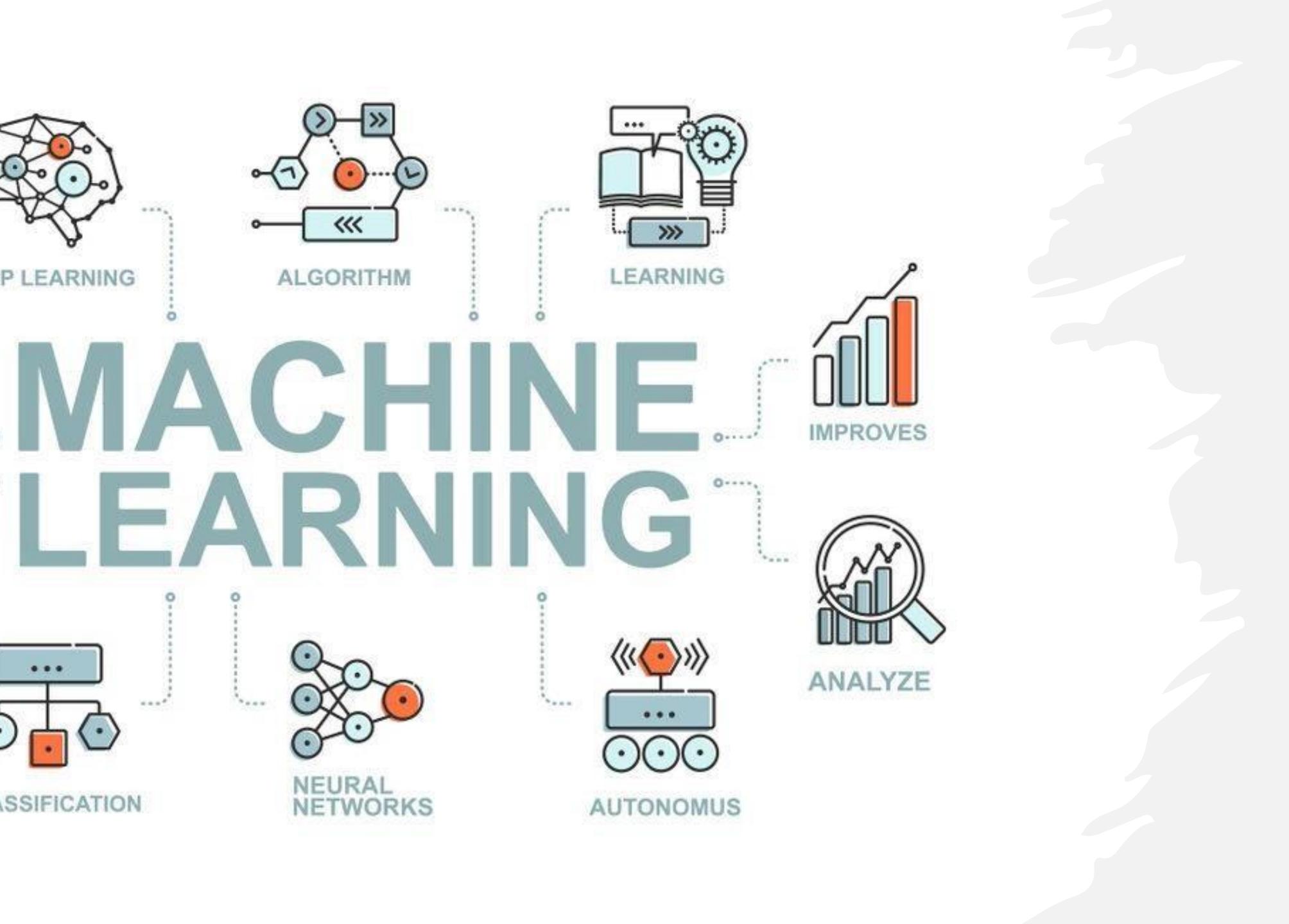
NEURAL NETWORKS



AUTONOMUS



ANALYZE



# 3 stages of AI



## Narrow AI

Dedicated to assist  
with or take over  
specific tasks



## General AI

Takes knowledge from  
one domain, transfers  
to other domain

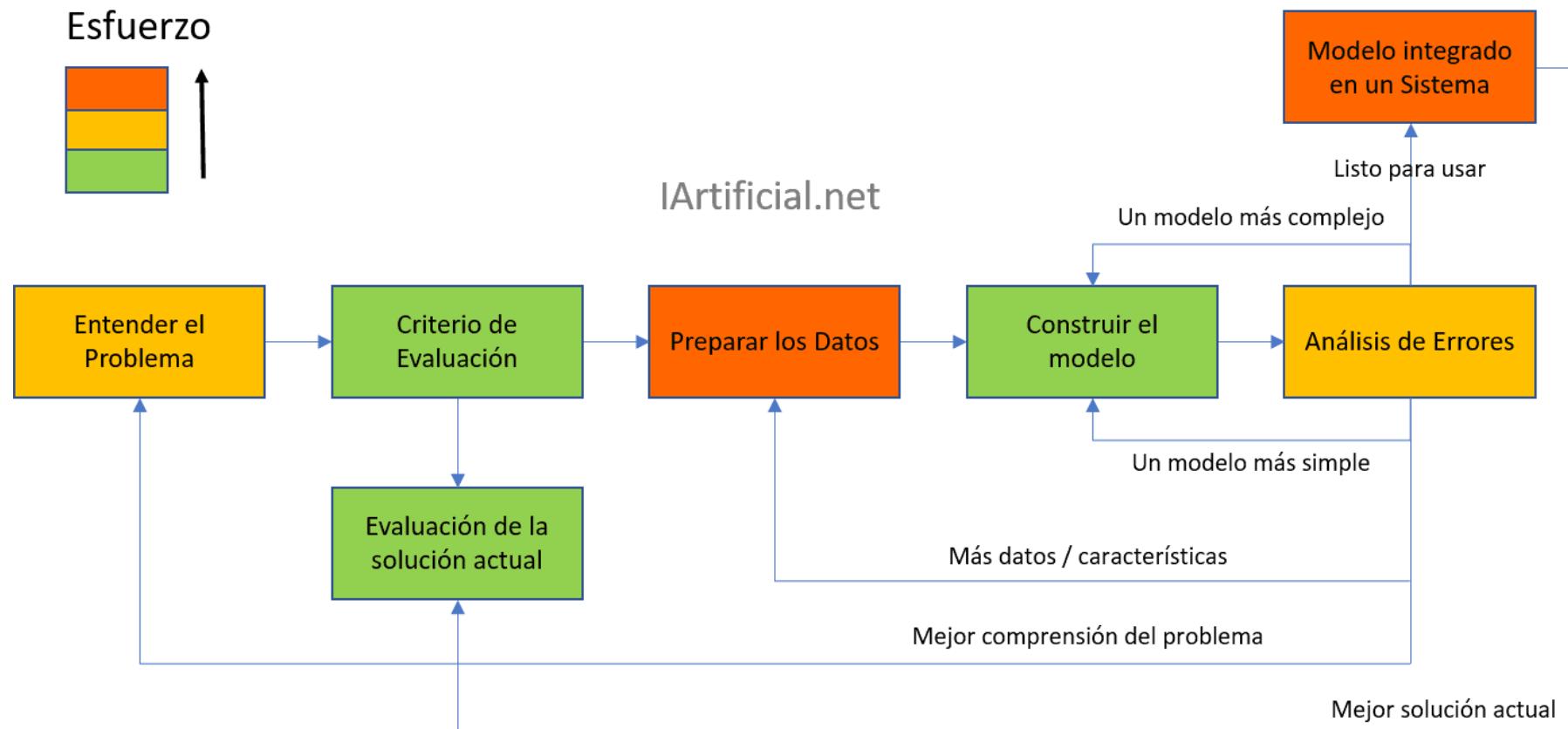


## Super AI

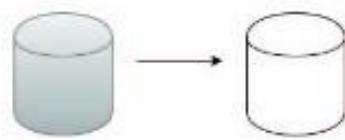
Machines that are an  
order of magnitude  
smarter than humans

Credit: Chris Noessel

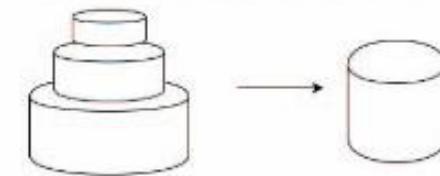
# PASOS DE UN PROYECTO DE ML



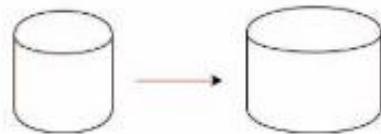
Limpieza de datos



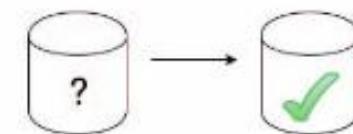
Normalización de datos



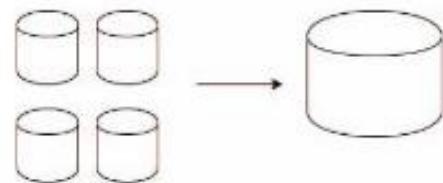
Transformación de datos



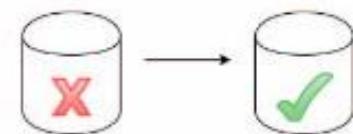
Imputación de valores perdidos



Integración de datos

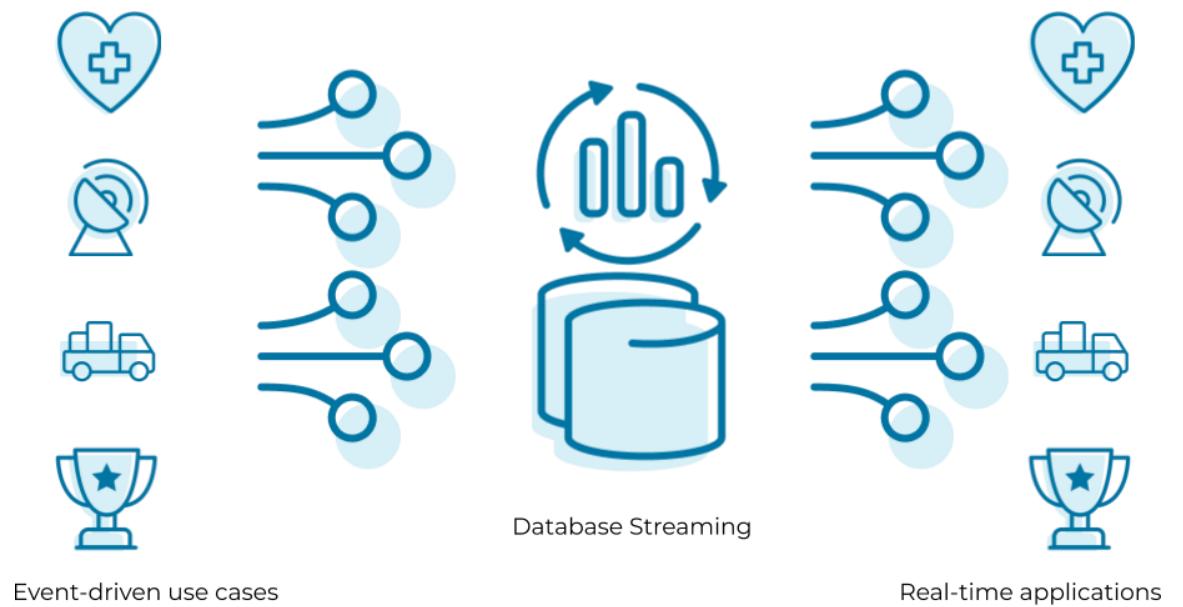


Identificación de ruido



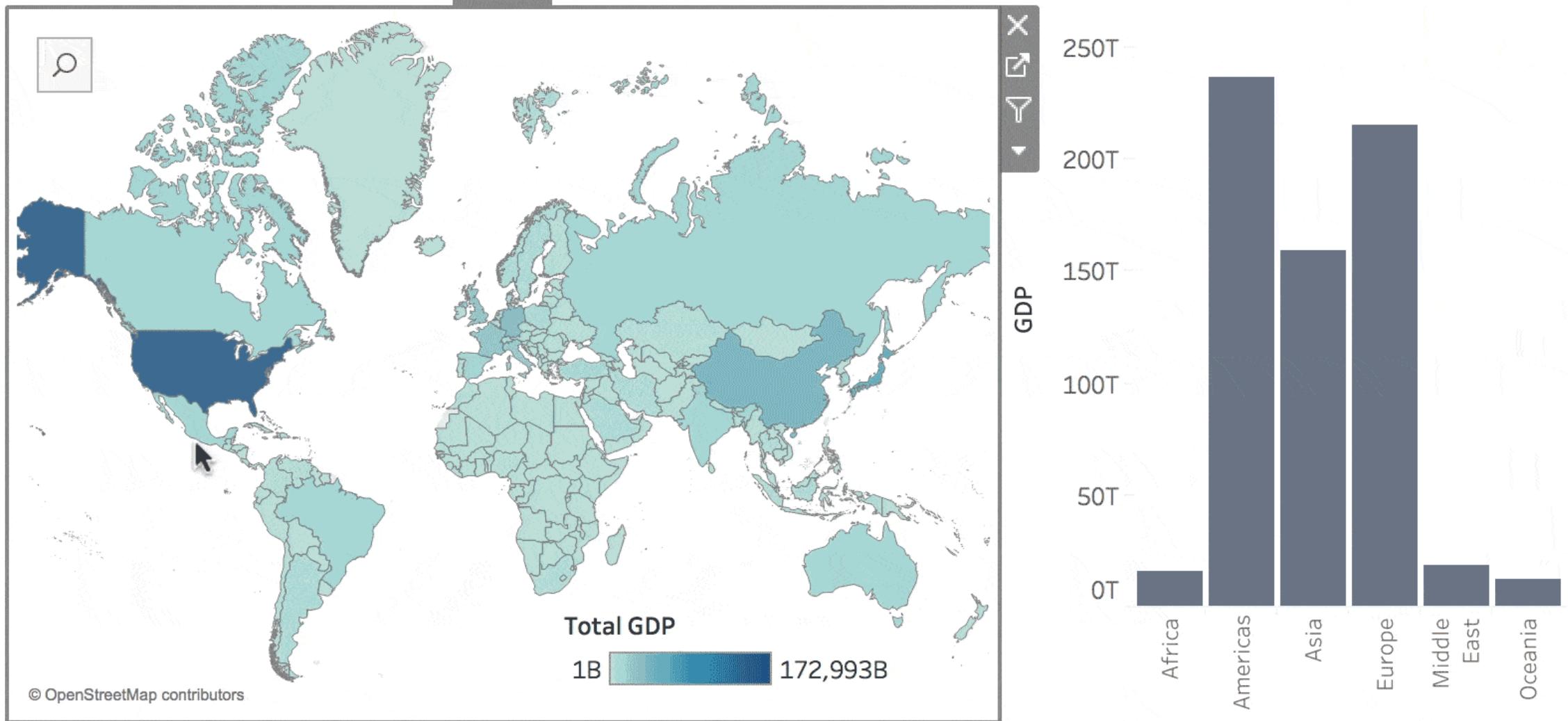
# PROBLEMAS EN LOS DATOS

# STREAMING DATA





# GDP by Country

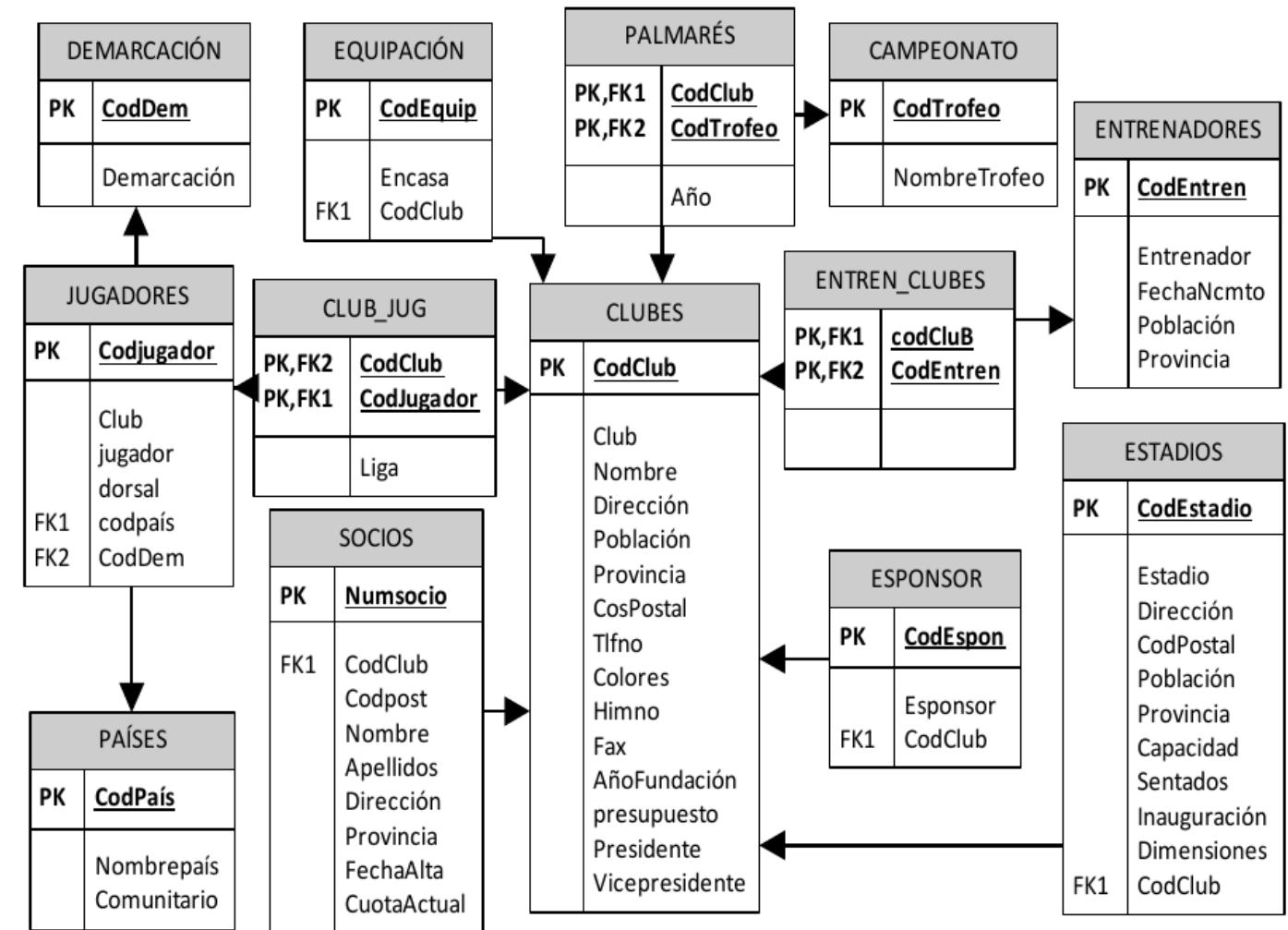


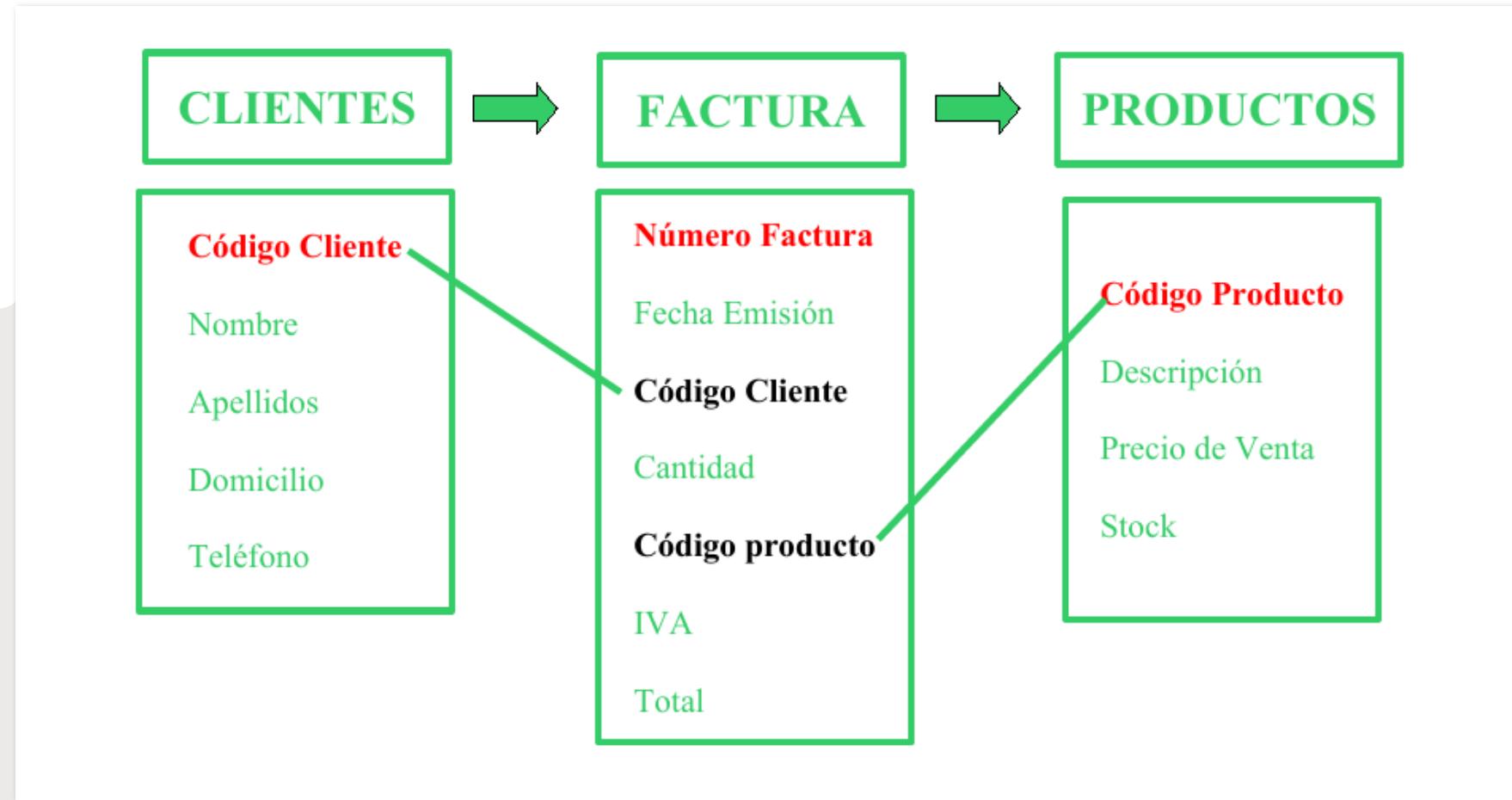
# BASES DE DATOS RELACIONALES Y SQL

# SISTEMA TRADICIONAL

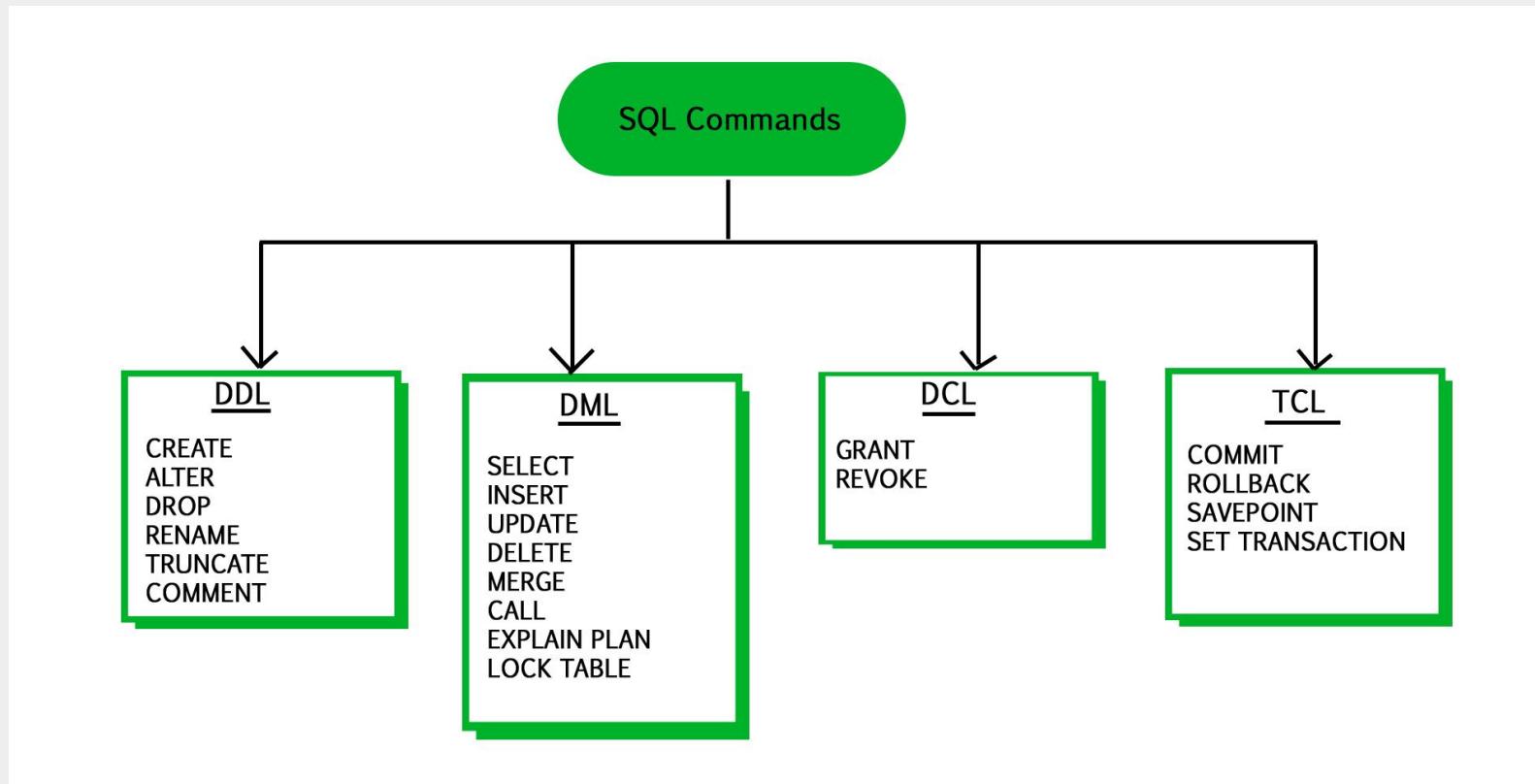
JUGADORES		PORTEROS		BARCELONA		REAL_MADRID		JUG_LIGA_PASADA		JUG_LIGA_ACTUAL	
PK	<u>Codjugador</u>	PK	<u>Codjugador</u>	PK	<u>Codjugador</u>	PK	<u>Codjugador</u>	PK	<u>Codjugador</u>	PK	<u>Codjugador</u>
	Club jugador dorsal codpaís codDem		club jugador dorsal CodPaís NombDem		Club jugador dorsal codpaís codDem		Club jugador dorsal codpaís codDem		Club jugador dorsal codpaís codDem		Club jugador dorsal codpaís codDem
CLUBES		CLUB_JUG		ESTADIOS		EQUIPACIÓN		SOCIOS		ESPOSOR	
PK	<u>CodClub</u>	PK	<u>CodJugador</u>	PK	<u>CodEstadio</u>	PK	<u>CodEquip</u>	PK	<u>Numsocio</u>	PK	<u>CodEspon</u>
Club Nombre Dirección Población Provincia CosPostal Tlfno Colores Himno Fax AñoFundación presupuesto Presidente Vicepresidente CodEquip Sponsor		CodClub		CodEquipo Estadio Dirección CodPostal Población Provincia Capacidad Sentados Inauguración Dimensiones		NombreEquip Encasa		CodClub Codpost Nombre Apellidos Dirección Provincia FechaAlta CuotaActual		CodClub Codpost Nombre Apellidos Dirección Provincia FechaAlta CuotaActual	
ENTRENADORES		PAÍSES		PALMARÉS		DEMARCIÓN					
PK	<u>CodEntren</u>	PK	<u>CodPaís</u>	PK	<u>CodClub</u>	PK	<u>CodTrofeo</u>	PK	<u>CodDem</u>	PK	<u>CodEspon</u>
ENTREN_CLUBES		CAMPEONATO		Año							
PK	<u>codCluB</u>	PK	<u>CodTrofeo</u>	PK	<u>CodDem</u>	PK	<u>CodEspon</u>	PK	<u>CodDem</u>	PK	<u>CodEspon</u>

# BBDD RELACIONAL





## RELACIONES



**Customers [-]**

customer_id [int]
first_name [varchar(100)]
last_name [varchar(100)]
age [int]
country [varchar(100)]

**Orders [-]**

order_id [integer]
item [varchar(100)]
amount [integer]
customer_id [integer]

**Shipments [-]**

shipping_id [integer]
status [integer]
customer [integer]

**Input**

```
-- Online SQL Editor to Run SQL Online.  
-- Use the editor to create new tables, insert data and all other SQL operations.  
  
SELECT first_name, age  
FROM Customers;
```

**Run SQL****Available Tables****Customers**

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

**Orders**

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

**Output**

first_name	age
John	31
Robert	22
David	22
John	25
Betty	28

**Shipments**

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT age, country  
FROM Customers;

age	country
31	USA
22	USA
22	UK
25	UK
28	UAE

# CLAUSULA SELECT

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT age, country  
FROM Customers  
WHERE age < 27;
```

age	country
22	USA
22	UK
25	UK

# CLAUSULA WHERE

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
=	equal to
<> or !=	not equal to

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT age, country  
FROM Customers  
WHERE country = 'USA';
```

age	country
31	USA
22	USA

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, age, country  
FROM Customers  
WHERE age > 23 AND last_name = 'Doe';
```

first_name	age	country
John	31	USA
Betty	28	UAE

**Table: Customers**

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT first_name, age, country  
FROM Customers  
WHERE age > 23 OR country = 'USA' ;
```

first_name	age	country
John	31	USA
Robert	22	USA
John	25	UK
Betty	28	UAE

```
1 -- select rows if country is not UK  
2  
3 SELECT *  
4 FROM Customers  
5 WHERE NOT country = 'UK';
```

NOT

```
1 -- select customers
2 -- if age > 23 and
3 -- country is either 'USA' or 'UK'
4
5 SELECT *
6 FROM Customers
7 WHERE age > 23 AND (country = 'USA' OR country = 'UK');
```

# COMBINADOS

**Table: Customers**

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

`SELECT DISTINCT country  
FROM Customers;`

country
USA
UK
UAE

```
-- select distinct countries  
-- then filter rows where age > 26
```

```
SELECT DISTINCT country  
FROM Customers  
WHERE age > 26;
```

DISTINCT + WHERE

**Table: Customers**

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
5	Betty	Doe	28	UAE
7	John	Doe	27	UAE
8	Robert	Luna	22	Canada

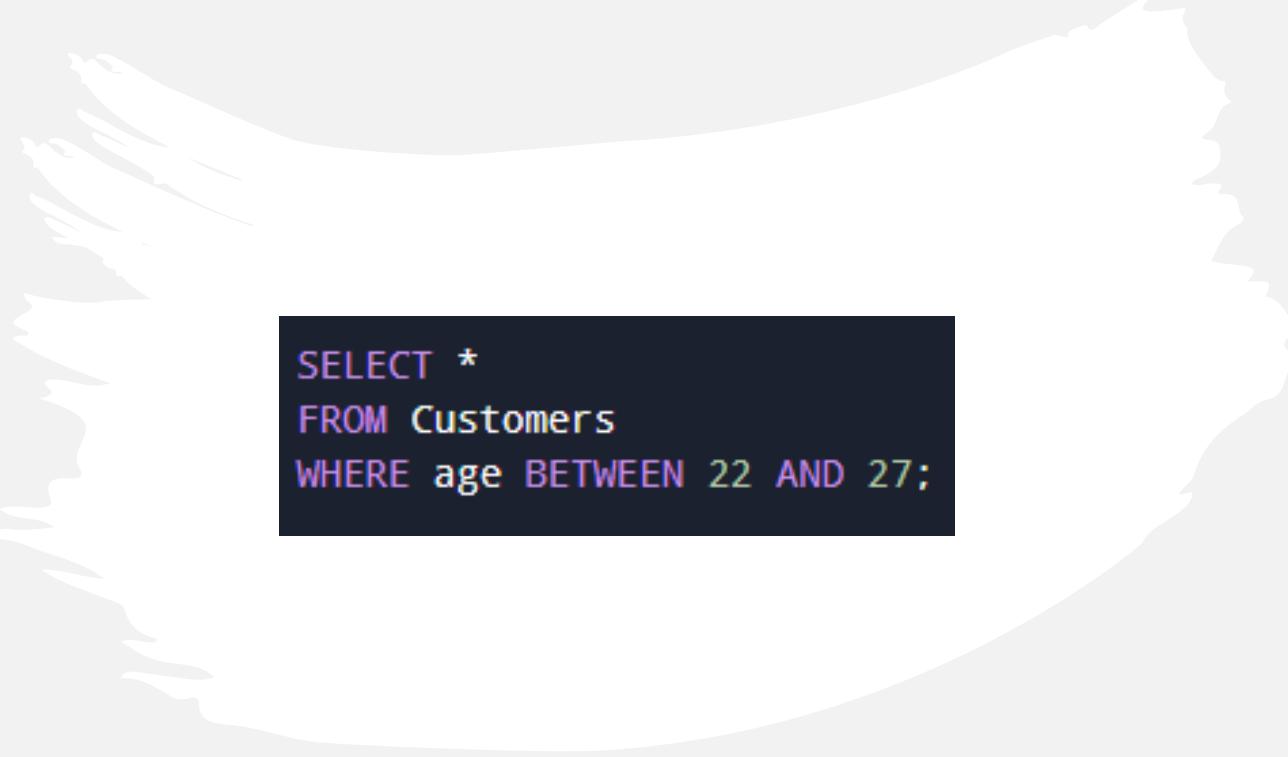
`SELECT DISTINCT first_name, last_name  
FROM Customers;`

first_name	last_name
John	Doe
Robert	Luna
Betty	Doe



```
SELECT first_name, country  
FROM Customers  
WHERE country IN ('USA', 'UK');
```

IN



```
SELECT *
FROM Customers
WHERE age BETWEEN 22 AND 27;
```

BETWEEN

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers
ORDER BY first_name ASC;
```

customer_id	first_name	last_name	age	country
5	Betty	Doe	28	UAE
3	David	Robinson	22	UK
1	John	Doe	31	USA
4	John	Reinhardt	25	UK
2	Robert	Luna	22	USA

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT *
FROM Customers
ORDER BY first_name DESC, age ASC;
```

customer_id	first_name	last_name	age	country
2	Robert	Luna	22	USA
4	John	Reinhardt	25	UK
1	John	Doe	31	USA
3	David	Robinson	22	UK
5	Betty	Doe	28	UAE

# LIMIT

```
-- select the first three rows  
  
SELECT *  
FROM Products  
LIMIT 2;
```

Table: Employees

id	first_name	last_name	department	age
1	Peter	Parker	Operations	25
2	Meghan	Markle	Finance	26
3	Joe	Rogan	Finance	27
4	Mike	Tyson	Sales	25
5	Mary	Beth	Operations	24
7	Elon	Gates	Sales	19
8	Samantha	Jones	Marketing	26

Table: Employees

id	first_name	last_name	department	age
1	Peter	Parker	Operations	25
2	Meghan	Markle	Finance	26
3	Joe	Rogan	Finance	27
4	Mike	Tyson	Sales	25
5	Mary	Beth	Operations	24
7	Elon	Gates	Sales	19
8	Samantha	Jones	Marketing	26

`SELECT MIN(age)  
FROM Employees;`

MIN(age)

19

`SELECT MAX(age)  
FROM Employees;`

MAX(age)

27

```
-- count the number of employees  
-- based on all columns  
  
SELECT COUNT(*)  
FROM Employees;
```

```
-- count the number of employees  
-- based on the last_name column  
  
SELECT COUNT(last_name)  
FROM Employees;
```

# COUNT

```
SELECT SUM(salary) AS total_sales_salary  
FROM Employees  
WHERE department = 'Sales';
```

```
SELECT AVG(age) AS USA_average_age  
FROM Customers  
WHERE country = 'USA';
```

SUM Y AVG

# GROUP BY

```
-- group data by different departments  
  
SELECT department, salary  
FROM Employees  
GROUP BY department;
```

### Ejemplo 1: Salario promedio de los empleados en cada departamento

```
1 SELECT department, AVG(salary) AS average_salary  
2 FROM Employees  
3 GROUP BY department;
```

[Ejecutar código >>](#)

La consulta anterior agrupa a los empleados por `department` y luego utiliza la `AVG()` función para calcular el salario promedio de cada departamento.

### Ejemplo 2: Número de empleados en cada departamento

```
1 SELECT department, COUNT(*) AS number_of_employees  
2 FROM Employees  
3 GROUP BY department;
```

[Ejecutar código >>](#)

### Ejemplo 3: Salarios máximos y mínimos de los empleados de cada departamento

```
SELECT department, MIN(salary) AS min_salary, MAX(salary) AS max_salary  
FROM Employees  
GROUP BY department;
```

[Ejecutar código >>](#)

```
SELECT department, AVG(salary)
FROM Employees
GROUP BY department
ORDER BY salary DESC;
```

```
SELECT department, AVG(salary)
FROM Employees
GROUP BY department
ORDER BY AVG(salary) DESC;
```

ORDENAR POR SALIDA GRUPOS

# VARIAS COLUMNAS

```
-- group data by customer_name  
-- and then by product  
  
SELECT customer_name, product, SUM(price * quantity) AS total_amount  
FROM Sales  
GROUP BY customer_name, product;
```

```
1 -- Syntax error: the WHERE clause cannot be used in this SQL query
2
3 SELECT department, AVG(salary) AS average_salary
4 FROM Employees
5 GROUP BY department
6 WHERE AVG(salary) > 4000;
```

Ejecutar código >>

Para filtrar registros en tales situaciones, utilizamos la `HAVING` cláusula.

```
1 -- group employees by departments
2 -- display records if the average salary of
3 -- employees in a department is more than 4000
4
5 SELECT department, AVG(salary) AS average_salary
6 FROM Employees
7 GROUP BY department
8 HAVING AVG(salary) > 4000;
```

Ejecutar código >>

Por cierto, puedes usar `WHERE` en una `GROUP BY` cláusula como esta:

```
1 SELECT department, SUM(salary) AS total_salary  
2 FROM Employees  
3 WHERE salary > 4000  
4 GROUP BY department;
```

[Ejecutar código >>](#)

Aquí, la consulta SQL primero selecciona filas si el salario es superior a **4000**. Luego, agrupa los resultados filtrados por `department`.

Sin embargo, no puedes usar `WHERE` en una `GROUP BY` cláusula como esta:

```
1 SELECT department, SUM(salary) AS total_salary  
2 FROM employees  
3 GROUP BY department  
4 WHERE salary > 4000;
```

[Ejecutar código >>](#)

En este caso, estamos intentando filtrar los resultados del `GROUP BY` uso de la `WHERE` cláusula. Esto da como resultado un error.

En su lugar, deberíamos usar la `HAVING` cláusula para filtrar los resultados de una `GROUP BY` cláusula.

```
1 -- select customers who are from USA  
2  
3 SELECT *  
4 FROM Customers  
5 WHERE country = 'USA';
```

Ejecutar código >>

Sin embargo, en lugar de filtrar filas según un país específico, ¿qué sucede si queremos filtrar filas que coinciden con un patrón, como clientes cuyo país comienza con la letra 'U'?

En tales casos, podemos utilizar el `LIKE` operador para seleccionar filas que coincidan con un patrón específico en una columna. Por ejemplo,

```
1 -- select customers whose country starts with the letter 'U'  
2  
3 SELECT *  
4 FROM Customers  
5 WHERE country LIKE 'U%';
```

Ejecutar código >>

Aquí, 'U%' indica cualquier cadena que comience con 'U'. El % comodín indica que 'U' puede ir seguido de cero o más caracteres.

Expression	String	Matched?
'R%	'R'	match
'R%'	'Run'	match
'R%'	'Summer'	no match
'%R'	'Aron'	no match
'%R'	'Summer'	match
'%R%	'Aron'	match
'%R%'	'Summer'	match
'%R%'	'Alu'	no match

Expression	String	Matched?
'R_'	'R'	no match
'R_'	'Ra'	match
'R_'	'Ran'	no match
'R_'	'Or'	no match
'_R'	'Or'	match
'_R'	'Number'	no match
'_R_'	'Ran'	no match
'_R_'	'Ara'	match
'__R_'	'Dara'	match
'__R_'	'Ara'	no match

**Customers**

customer_id	customer_name	age	country
1	Peter Smith	22	USA
2	Meghan Burns	21	UK
3	Elon Gates	27	USA
4	John Doe	27	USA
5	Betty Doe	23	UK

**Orders**

order_id	customer_id	item	amount
1	1	mouse	300
2	1	keyboard	400
3	3	mouse	300
4	4	mouse	300

**Ecommerce**

customer_id	customer_name	age	country	item	amount
1	Peter Smith	22	USA	mouse	300
1	Peter Smith	22	USA	keyboard	400
2	Meghan Burns	21	UK		
3	Elon Gates	27	USA	mouse	300
4	John Doe	27	USA	mouse	300
5	Betty Doe	23	UK		

# JOINS

# INNER JOIN

**SQL INNER JOIN**

Table: Customers

customer_id	name	city
1	John	New York
2	Robert	Los Angeles
3	David	Chicago
4	Jane	New York

Table: Orders

order_id	customer_id	product
1	1	Computer
3	2	Phone
4	1	Tablet
6	3	Phone
7	3	Computer

A large blue bracket on the left side of the diagram groups both the Customers and Orders tables, indicating they are being joined together.

A blue arrow points from the bottom of the joined table to the resulting table below:

name	city	product
John	New York	Computer
Robert	Los Angeles	Phone
John	New York	Tablet
David	Chicago	Phone
David	Chicago	Computer

Customers

customer_id	name	city
1	John	New York
2	Robert	Los Angeles
3	David	Chicago
4	Jane	New York

Orders

order_id	customer_id	product
1	1	Computer
3	2	Phone
4	1	Tablet
6	3	Phone
7	3	Computer

select these columns

SELECT Customers.name, Customers.city, Orders.product

FROM Customers

join these tables

INNER JOIN Orders

ON Customers.customer\_id = Orders.customer\_id;

join tables on this condition

## Bad Practice

```
SELECT name, product  
FROM Customers c  
INNER JOIN Orders o  
ON c.customer_id = o.customer_id  
WHERE product = 'Computer';
```

## Good Practice

```
SELECT c.name, o.product  
FROM Customers c  
INNER JOIN Orders o  
ON c.customer_id = o.customer_id  
WHERE o.product = 'Computer';
```

**Customers**

customer_id	name	city
1	John	New York
2	Robert	Los Angeles
3	David	Chicago
4	Jane	New York

**Orders**

order_id	customer_id	product_id
1	1	2
2	2	3
3	1	1
5	4	1
6	1	3

**Products**

product_id	product	amount
1	Computer	1200
2	Phone	999
3	Tablet	650

## SQL LEFT JOIN

Table: Customers

customer_id	name	city
1	John	New York
2	Robert	Los Angeles
3	David	Chicago
4	Jane	New York

Table: Orders

order_id	customer_id	product
1	1	Computer
4	1	Tablet
6	3	Phone
7	3	Computer

customer_id	name	city	order_id	customer_id	product
1	John	New York	1	1	Computer
1	John	New York	4	1	Tablet
2	Robert	Los Angeles			
3	David	Chicago	6	3	Phone
3	David	Chicago	7	3	Computer
4	Jane	New York			

SELECT \*

FROM Customers left table  
LEFT JOIN Orders right table

ON Customers.customer\_id = Orders.customer\_id;

## SQL RIGHT JOIN

Table: Customers

customer_id	name	city
1	John	New York
2	Robert	Los Angeles
3	David	Chicago
4	Jane	New York

Table: Orders

order_id	customer_id	product
1	1	Computer
4	1	Table
6	3	Phone
7	5	Computer

customer_id	name	city	order_id	customer_id	product
1	John	New York	1	1	Computer
1	John	New York	4	1	Table
3	David	Chicago	6	3	Phone
			7	5	Computer

### WORKING OF DIFFERENT JOINS

History_Grades			Math_Grades		
student_id	student_name	history_grade	student_id	student_name	math_grade
1	Alice	A	1	Alice	B
2	Jane	B	4	Roger	A
3	Julie	A	5	Kate	C

#### INNER JOIN

student_id	student_name	history_grade	student_id	student_name	math_grade
1	Alice	A	1	Alice	B

#### LEFT JOIN

student_id	student_name	history_grade	student_id	student_name	math_grade
1	Alice	A	1	Alice	B
2	Jane	B			
3	Julie	A			

#### RIGHT JOIN

student_id	student_name	history_grade	student_id	student_name	math_grade
1	Alice	A	1	Alice	B
			4	Roger	A
			5	Kate	C

#### FULL JOIN

student_id	student_name	history_grade	student_id	student_name	math_grade
1	Alice	A	1	Alice	B
2	Jane	B			
3	Julie	A	4	Roger	A
			5	Kate	C

# INSERT INTO

Students

<b>id</b>	<b>name</b>	<b>age</b>
1	Harry	23
2	Jack	45
3	Joe	32
4	Dent	23
5	Bruce	40
8	Jules	22

**INSERT INTO**  
table name → **Students** (**id, name, age**)  
**VALUES** → **(8, 'Jules', 22);**  
                  |        |  
                  |        | → column names  
                  |        |  
                  |        | → values to be inserted into  
                  |        | id, name, and age respectively

## Products

<b>id</b>	<b>name</b>	<b>price</b>	<b>quantity</b>
1	keyboard	250	25
2	mouse	175	22
7	headphone	150	20
	HDMI Cable	15	

← This row is inserted.

**Table: Customers**

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
UPDATE Customers  
SET first_name = 'Johnny';
```

**Table: Customers**

customer_id	first_name	last_name	age	country
1	Johnny	Doe	31	USA
2	Johnny	Luna	22	USA
3	Johnny	Robinson	22	UK
4	Johnny	Reinhardt	25	UK
5	Johnny	Doe	28	UAE

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
UPDATE Customers  
SET first_name = 'Johnny'  
WHERE customer_id = 1;
```

update the **first\_name**  
column for row with  
**customer\_id** of 1

Table: Customers

customer_id	first_name	last_name	age	country
1	Johnny	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

# VARIAS COLUMNAS

```
UPDATE Customers
SET first_name = 'Johnny', last_name = 'Burns'
WHERE customer_id = 1;
```

**Table: Customers**

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
DELETE FROM Customers  
WHERE country = 'UK';
```

**Table: Customers**

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
5	Betty	Doe	28	UAE

# CREAR TABLAS

```
CREATE TABLE Companies (
    id int,
    name varchar(100),
    email varchar(50),
    phone varchar(10)
);
```

keyword ←  
CREATE TABLE Companies ( ← table name  
column name → id int , → data type  
column name → name varchar(100) , → data type  
email varchar(50),  
phone varchar(10)  
);

```
CREATE TABLE Companies (
    id int,
    name varchar(100),
    email varchar(50),
    phone varchar(10)
);
```

-- change a column's data type

```
ALTER TABLE Products  
MODIFY COLUMN name TEXT;
```

-- rename a column

```
ALTER TABLE Products  
RENAME COLUMN price TO amount;
```

-- add a column

```
ALTER TABLE Products  
ADD COLUMN date_added DATE;
```

-- delete a column

```
ALTER TABLE Products  
DROP COLUMN quantity;
```

# ALTER TABLE

```
-- create table
-- the id column has the NOT NULL constraint

CREATE TABLE Customers (
    id INT NOT NULL,
    name VARCHAR(50),
    email VARCHAR(50)
);

-- insert data without providing value for the id column
INSERT INTO
    Customers(name, email)
VALUES
    ('Jack', 'jack@example.com');
```

Restricción	Descripción
NOT NULL	los valores no pueden ser NULL
UNIQUE	los valores no pueden coincidir con ningún valor anterior
PRIMARY KEY	utilizado para identificar de forma única una fila
FOREIGN KEY	hace referencia a una fila en otra tabla

# UNIQUE

```
-- create table
-- the id column has the UNIQUE constraint

CREATE TABLE Customers (
    id INTEGER UNIQUE,
    name VARCHAR(50),
    email VARCHAR(50)
);

-- insert a row
INSERT INTO Customers(id, name, email)
VALUES (1, 'Jack', 'jack@example.com');

-- insert another row
-- results in an error because the id column already contains 1

INSERT INTO Customers(id, name, email)
VALUES (1, 'Anita', 'anita@example.com');
```

A continuación se muestra un ejemplo para crear una tabla con una PRIMARY KEY restricción:

```
1 -- create a table
2 -- the id column has the PRIMARY KEY constraint
3 CREATE TABLE Customers (
4     id INTEGER PRIMARY KEY,
5     name VARCHAR(50),
6     email VARCHAR(50)
7 );
8
9 -- insert a row
10 INSERT INTO Customers(id, name, email)
11 VALUES (1, 'Jack', 'jack@example.com');
12
13 -- insert another row
14 -- results in an error because the id column already contains 1
15
16 INSERT INTO Customers(id, name, email)
17 VALUES (1, 'Anita', 'anita@example.com');
```

Ejecutar código >>

Cuando ejecutas el código,

1. `Customers` Se crea una tabla denominada .
2. Se inserta una fila en la tabla debido a la primera `INSERT INTO` declaración.
3. La segunda `INSERT INTO` declaración produce un error porque estamos intentando insertar 1 en la `id` columna nuevamente.

```
SELECT *, test condition  
  CASE  
    WHEN age >= 20 AND age <= 25 THEN '20-25'  
  END AS age_group  
FROM Customers;
```

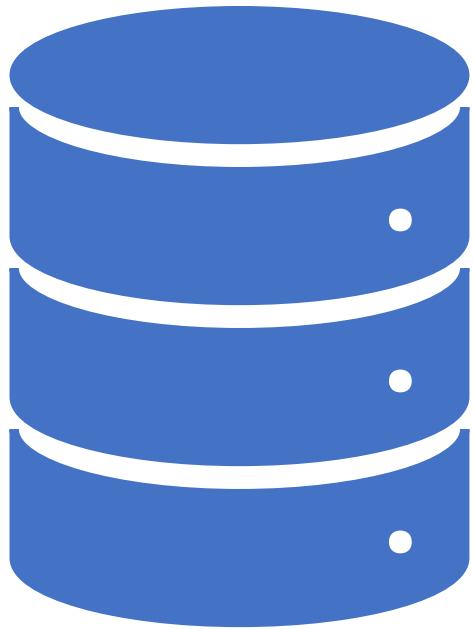
new column name  
in the output

value if the test  
condition is TRUE



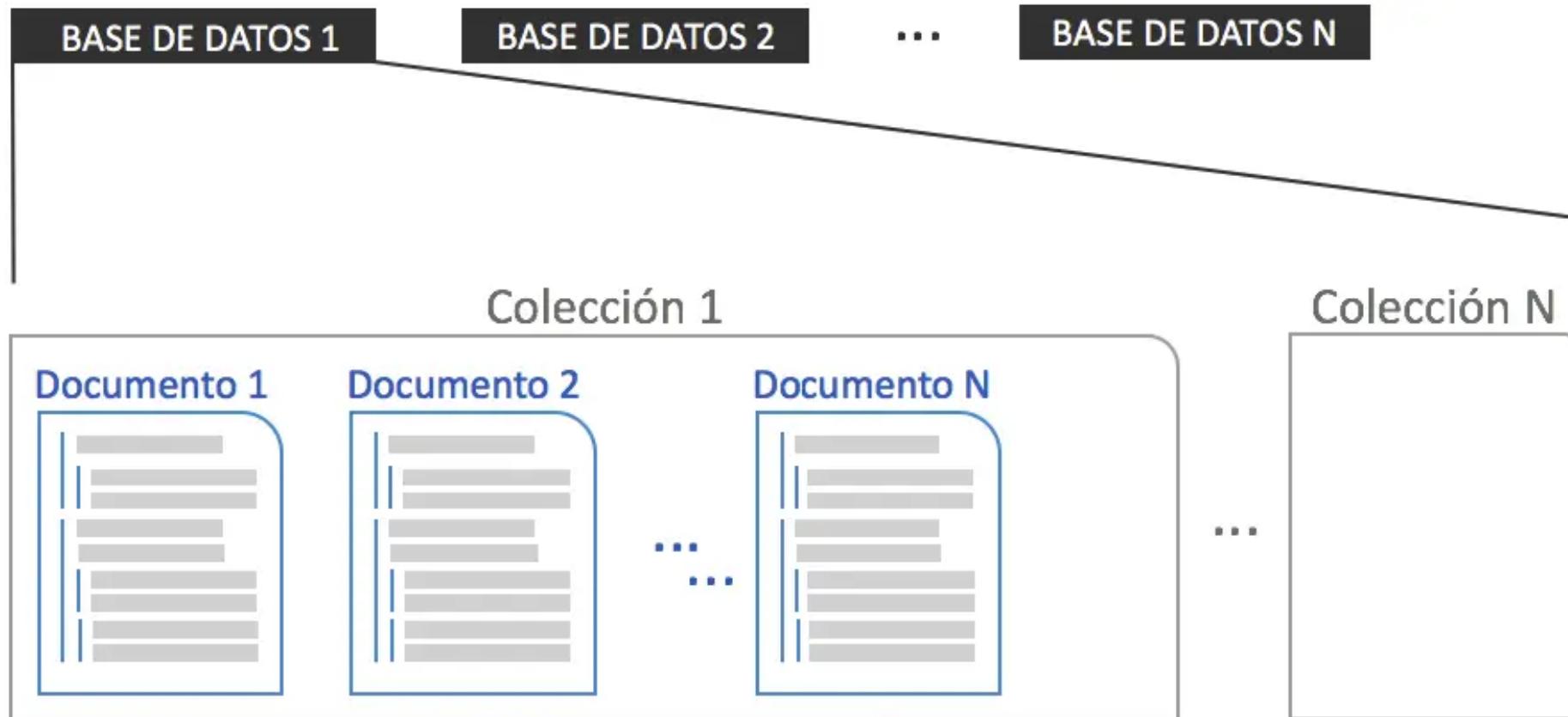
MongoDB®

# MONGODB



- El gestor de base de datos MongoDB se lo puede asociar a un conjunto de gestores de bases de datos que no tienen como lenguaje principal el SQL para su manipulación.
- Los gestores de bases de datos NoSQL no requieren estructuras fijas como tablas, normalmente no soportan operaciones join y presentan como gran ventaja que pueden escalar en forma sencilla.

# BASE DE DATOS DOCUMENTAL





MOTORES NOSQL



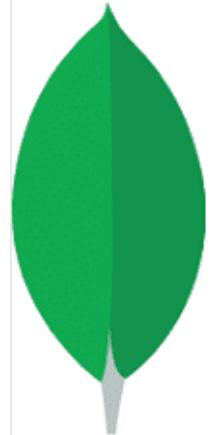
# CARACTERÍSTICAS MONGODB

- Indexación
- Replicación
- Balanceo de carga
- Almacenamiento de archivos
- Agregación



# JSON

```
{  
  codigo: 1,  
  nombre: 'El aleph',  
  autor: 'Borges',  
  editoriales: ['Planeta','Siglo XXI']  
}
```



mongoDB® Atlas

# MONGODB COMPASS

localhost:27017  
Community version 3.1.8

8 DBs | 15 Collections | C

flightStats-cut

Query returned 9,993 documents. This report is based on a sample of 100 documents (1.00%). ⓘ

DOCUMENTS 10.0k total size 6.5 MB avg. size 684 B | INDEXES 1 total size 566.9 KB avg. size 566.9 KB

APPLY RESET

**\_id**  
string

EWR-EV-4467-542273341 EWR-EV-4382-544626614  
LGA-9E-3457-542758157 JFK-9E-4093-544640472  
LGA-ZW-3910-545111616 JFK-SE-41-544640167  
EWR-YX-4904-544626872 LGA-ZW-3815-544184788

**arrivalAirportF** String (100%)  
string

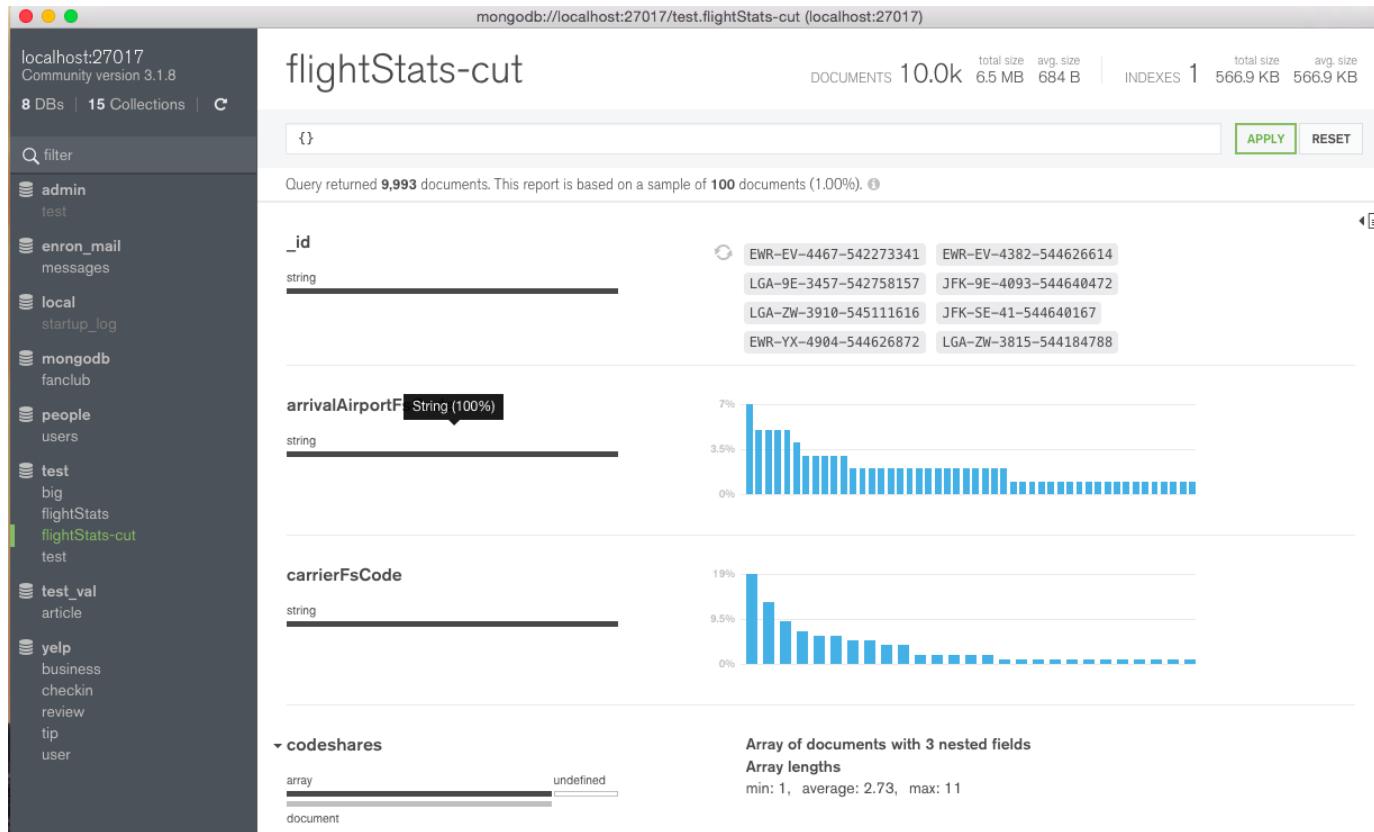
0% 3.5% 7%

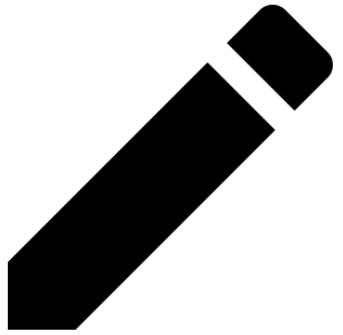
**carrierFsCode**  
string

0% 9.5% 19%

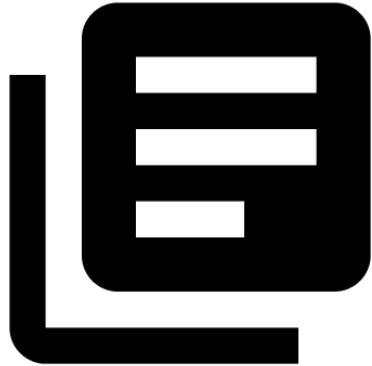
**codeshares**  
array undefined  
document

Array of documents with 3 nested fields  
Array lengths  
min: 1, average: 2.73, max: 11

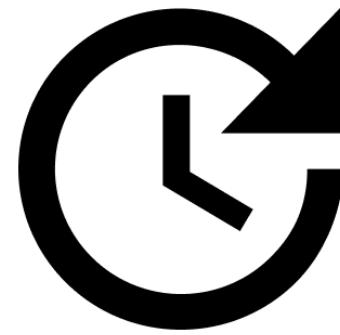




C  
reate



R  
ead



U  
pdate



D  
elete

A close-up photograph of a pair of dark-rimmed glasses resting on an open, lined notebook. A red ribbon or piece of tape holds the pages of the notebook together. The background is blurred, showing what appears to be a window with a view of a city skyline.

# INSERTAR DOCUMENTOS

Para insertar un documento o un conjunto de documentos disponemos de los métodos:

- `insertOne`: Inserta un documento en una colección.
- `insertMany`: Inserta múltiples documentos en una colección.

# CAMPO OBLIGATORIO \_ID

- En MongoDB, cada documento almacenado en una colección requiere un único `_id` que actúa como clave principal . Si se inserta documento omite el `_id`, el controlador MongoDB automáticamente genera un `ObjectId` para el `_id`.

```
1 > db.student.find( {} )  
2           collection name      empty query document  
3 |
```

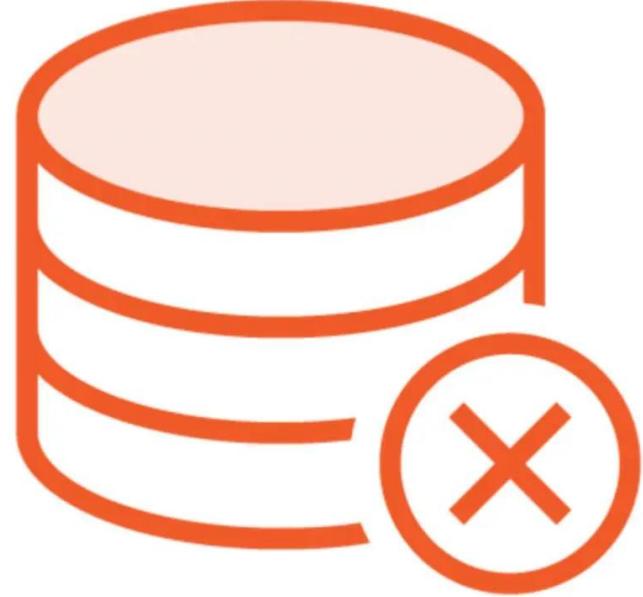
RECUPERAR DOCUMENTOS

# OPERADORES COMPARACIÓN

- \$eq - equal - igual
- \$lt - low than - menor que
- \$lte - low than equal - menor o igual que
- \$gt - greater than - mayor que
- \$gte - greater than equal - mayor o igual que
- \$ne - not equal - distinto
- \$in - in - dentro de
- \$nin - not in - no dentro de



# EJERCICIO 1



**db.collection.deleteOne()**  
**db.collection.deleteMany()**  
**db.collection.remove()**

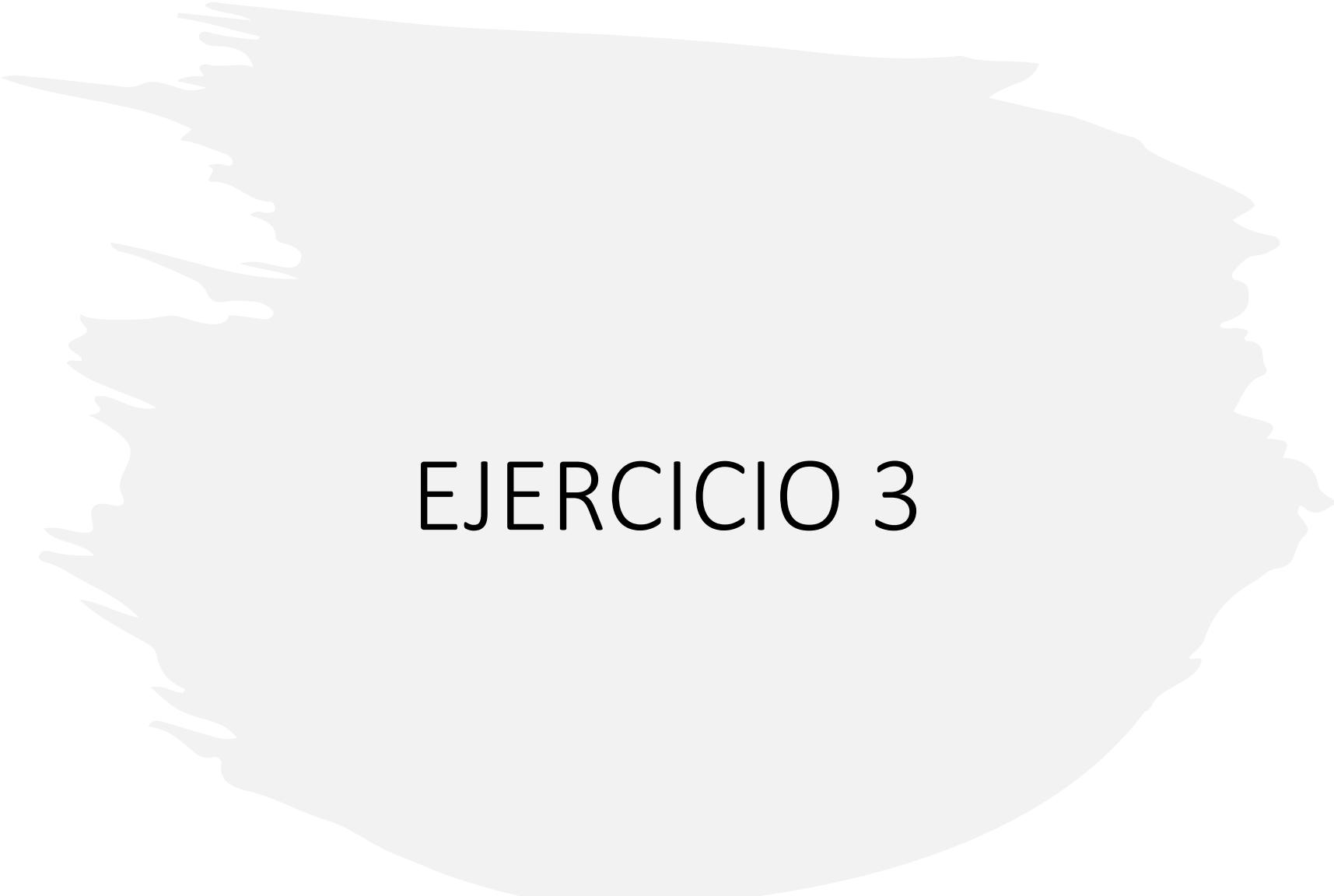
ELIMINAR DOCUMENTOS Y BBDD



## EJERCICIO 2

# MODIFICAR UN ELEMENTO

```
> db.job.updateOne({salary:5000}, { $set: {firstName:'Morgan'}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.job.find().pretty()
{
    "_id" : ObjectId("6286f50681063f5a55d4775b"),
    "firstName" : "Morgan",
    "lastName" : "Dew",
    "email" : "john.dew@abc.com",
    "salary" : 5000
}
```



## EJERCICIO 3

# MODIFICAR MÚLTIPLES ELEMENTOS

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```

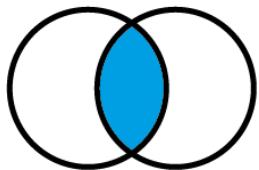
← collection  
← update filter  
← update action



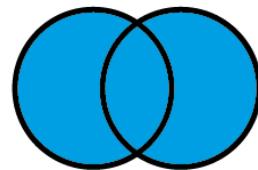
## EJERCICIO 4

# OPERADORES LÓGICOS

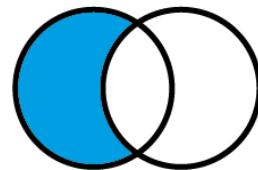
AND



OR



NOT





## EJERCICIO 5

# CURSORES Y MÉTODOS

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
...
...
...
}
...
}
{
  "acknowledged" : true, "insertedId" : 4
}
>
> db.libros.find().sort({titulo:1})
{
  "_id" : 3, "titulo" : "Aprenda PHP", "autor" : "Mario Molina", "editorial" : [ "Siglo XXI", "Planeta" ], "precio" : 50, "cantidad" : 20
}
{
  "_id" : 1, "titulo" : "El aleph", "autor" : "Borges", "editorial" : [ "Siglo XXI", "Planeta" ], "precio" : 20, "cantidad" : 50
}
{
  "_id" : 4, "titulo" : "Java en 10 minutos", "autor" : null, "editorial" : [ "Siglo XXI" ], "precio" : 45, "cantidad" : 1
}
{
  "_id" : 2, "titulo" : "Martin Fierro", "autor" : "Jose Hernandez", "editorial" : [ "Siglo XXI" ], "precio" : 50, "cantidad" : 12
}
>
```

# RECUPERAR SOLO ALGUNOS CAMPOS

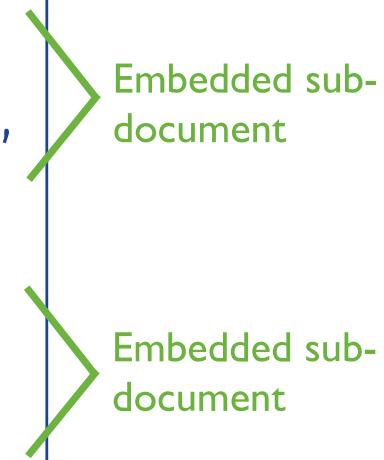
- Hemos visto que el método 'find':
  - Si no le pasamos parámetros nos retorna todos los documentos de la colección que hace referencia: db.libros.find({precio: 50},{titulo:1,cantidad:1,\_id:0})
  - El primer parámetro en el caso que lo indiquemos filtra la colección y recupera los documentos que cumplen la condición
  - En el segundo parámetro del método 'find' debemos especificar cada campo y un valor 1 indicando que se lo quiere recuperar.



## EJERCICIO 6

# DOCUMENTOS EMBEBIDOS

```
{  
  _id: <ObjectId1>,  
  username: "123xyz",  
  contact: {  
    phone: "123-456-7890",  
    email: "xyz@example.com"  
  },  
  access: {  
    level: 5,  
    group: "dev"  
  }  
}
```



Embedded sub-document

Embedded sub-document



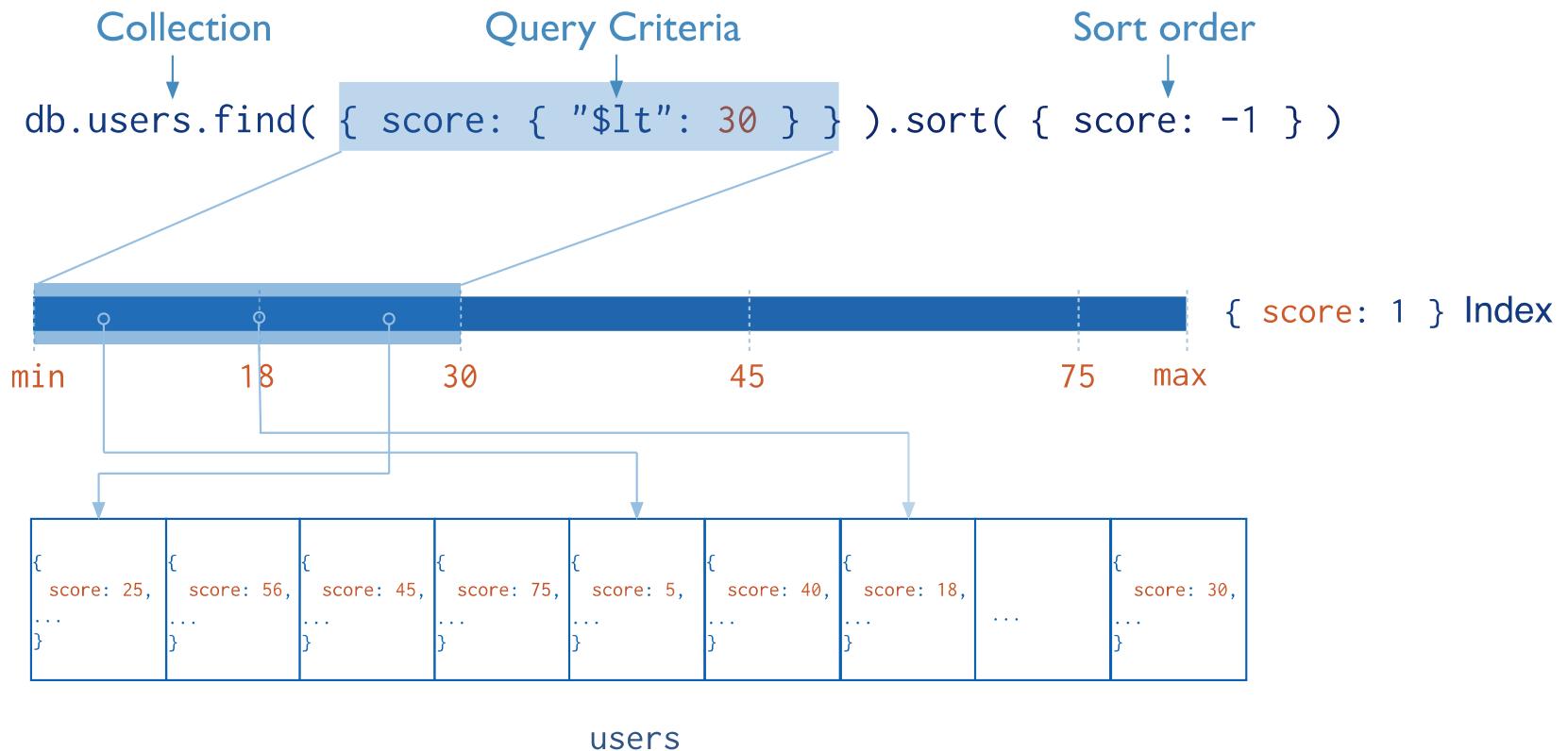
## EJERCICIO 7

```
C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe
>
> db.autos.find().pretty()
{
    "_id" : ObjectId("5c3a60d52c419f7680420488"),
    "patente" : "aaa111",
    "fechahora" : ISODate("2019-01-12T21:49:09.852Z")
}
{
    "_id" : ObjectId("5c3a60d52c419f7680420489"),
    "patente" : "bbb222",
    "fechahora" : ISODate("2019-01-12T21:49:09.873Z")
}
{
    "_id" : ObjectId("5c3a60d52c419f768042048a"),
    "patente" : "ccc333",
    "fechahora" : ISODate("2019-01-12T21:49:09.883Z")
}
>
```

DATE

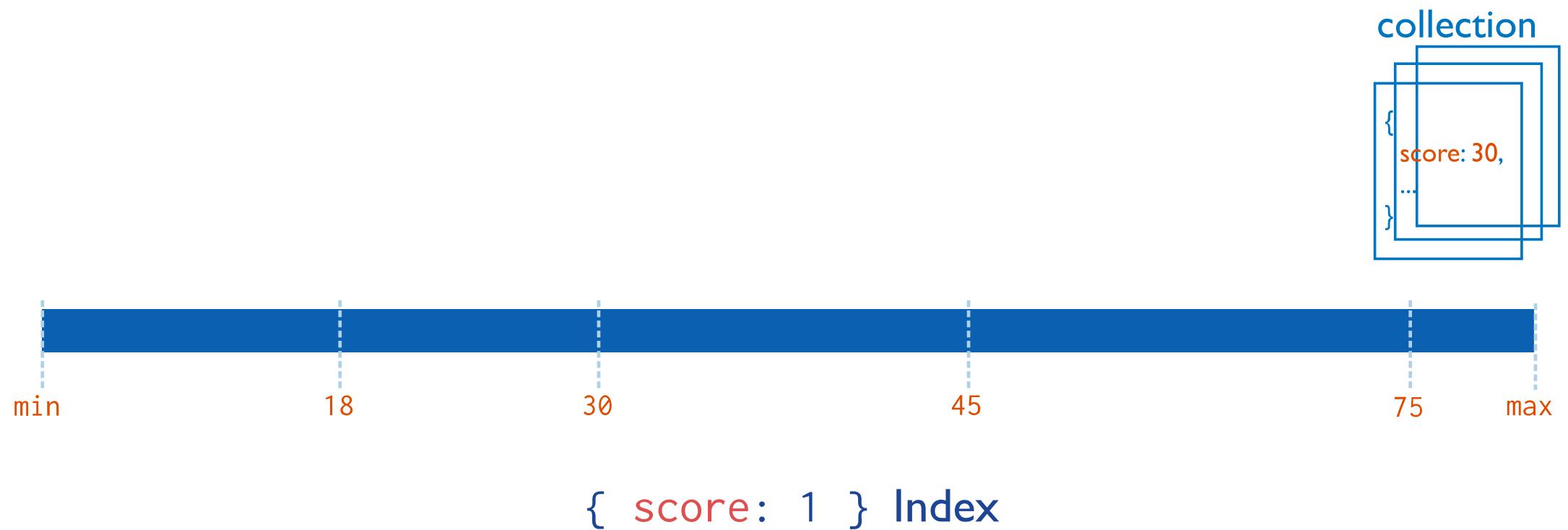


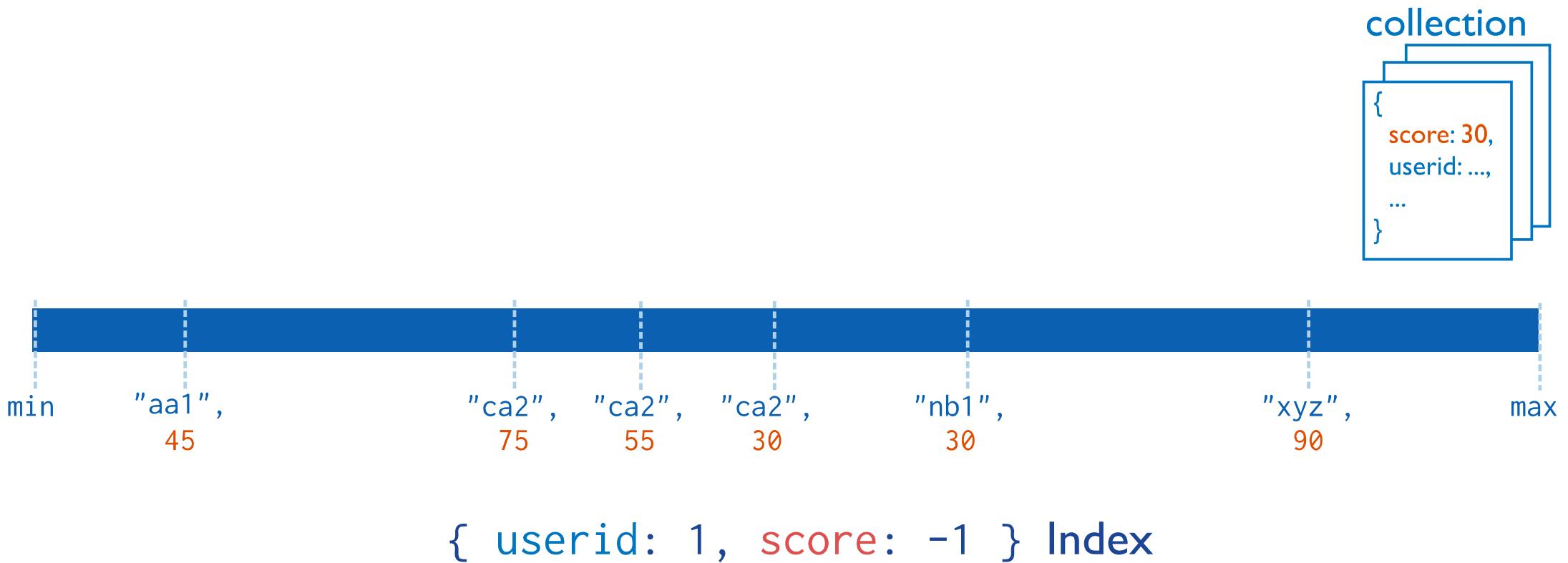
## EJERCICIO 8



# ÍNDICES

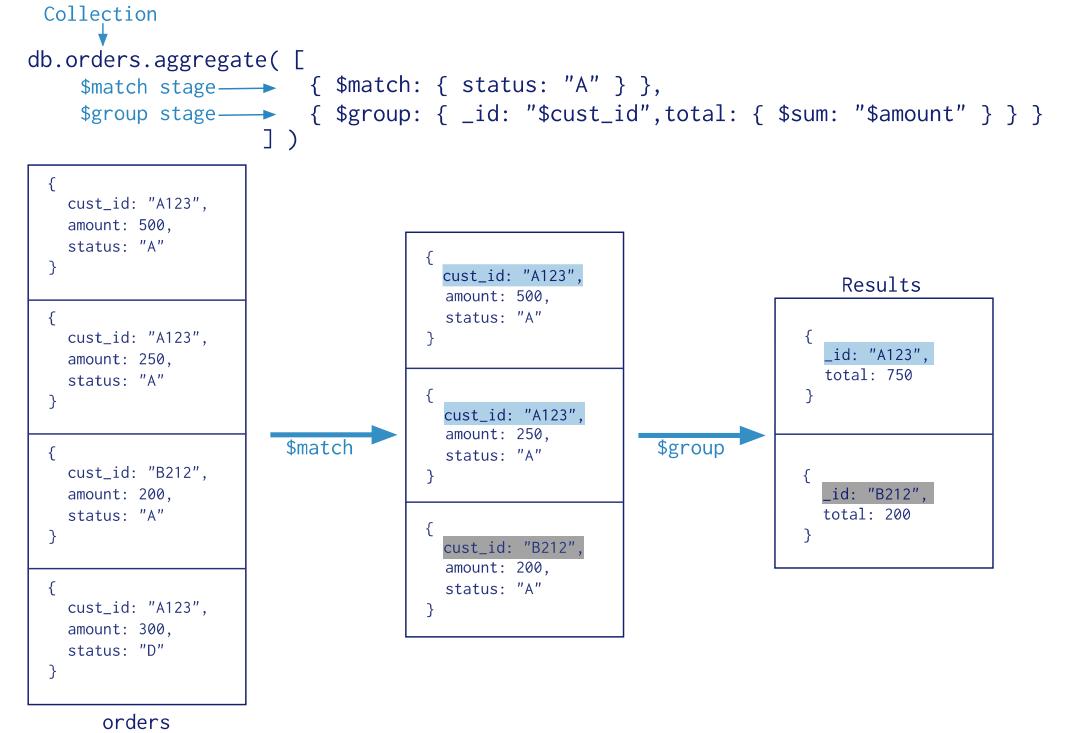
# ÍNDICE SIMPLE





ÍNDICE COMPUESTO

# AGGREGATION PIPELINE



# \$match

```
db.universities.aggregate([  
  { $match : { country : 'Spain', city : 'Salamanca' } }  
]).pretty()
```

# \$project

```
db.universities.aggregate([  
  { $project : { _id : 0, country : 1, city : 1, name : 1 } }  
]).pretty()
```

# \$group

```
db.universities.aggregate([
  { $group : { _id : '$name', totaldocs : { $sum : 1 } } }
]).pretty()
```

# \$out

```
db.universities.aggregate([
  { $group : { _id : '$name', totaldocs : { $sum : 1 } } },
  { $out : 'aggResults' }
])
```

# \$unwind

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $unwind : '$students' }
]).pretty()
```

# \$sort

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $unwind : '$students' },
  { $project : { _id : 0, 'students.year' : 1, 'students.number' : 1 } },
  { $sort : { 'students.number' : -1 } }
]).pretty()
```

# \$limit

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $unwind : '$students' },
  { $project : { _id : 0, 'students.year' : 1, 'students.number' : 1 } },
  { $sort : { 'students.number' : -1 } },
  { $limit : 2 }
]).pretty()
```

# \$addFields

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $addFields : { foundation_year : 1218 } }
]).pretty()
```

# \$count

```
db.universities.aggregate([
  { $unwind : '$students' },
  { $count : 'total_documents' }
]).pretty()
```

# \$lookup

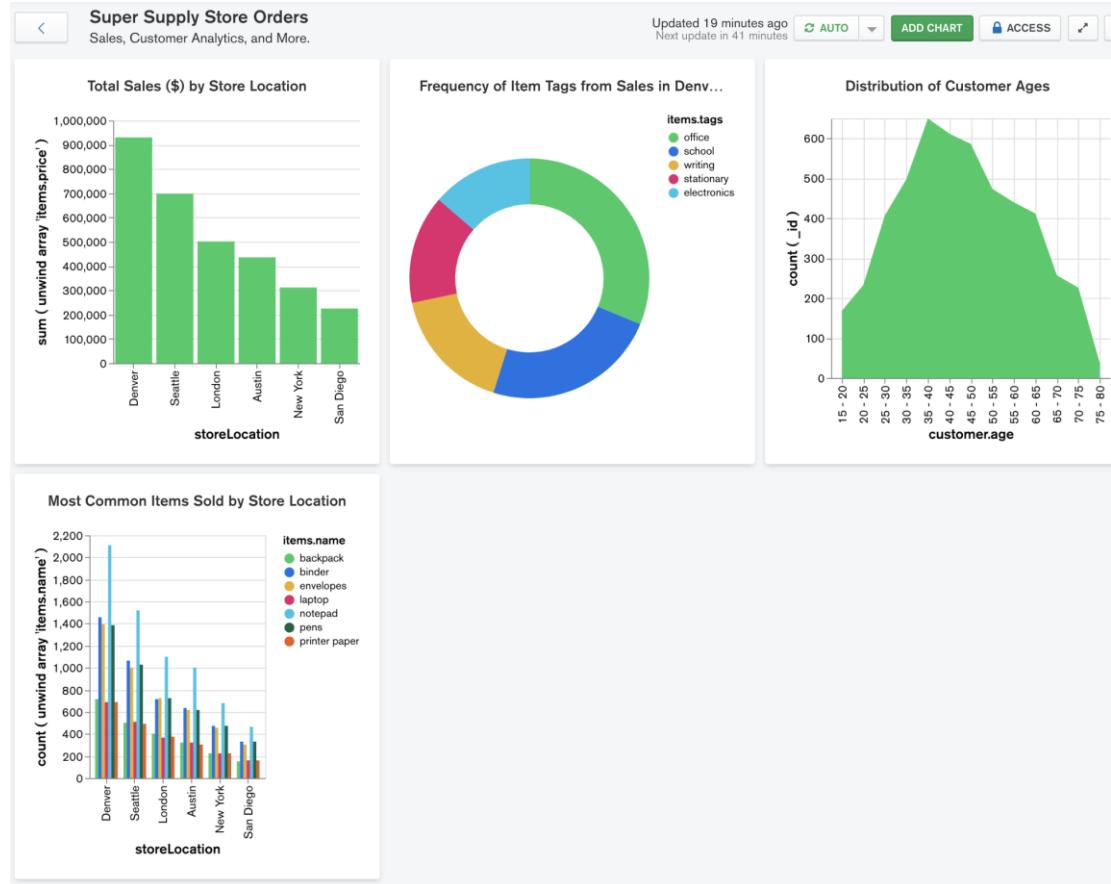
```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $project : { _id : 0, name : 1 } },
  { $lookup : {
    from : 'courses',
    localField : 'name',
    foreignField : 'university',
    as : 'courses'
  } }
]).pretty()
```

# `$sortByCount`

```
db.courses.aggregate([  
  { $sortByCount : '$level' }  
]).pretty()
```



# EJERCICIO 9



# CHARTS

# EJERCICIO 10

<https://www.mongodb.com/docs/charts/tutorial/movie-details/movie-details-tutorial-overview/>

¿QUÉ ES  
POWER BI?



Power BI Desktop

# ¿QUÉ ES BI?



# REVOLUCIÓN INDUSTRIAL



# COMPETIDORES





Power BI

vs



Tableau

vs



Qlik

# POWER PLATFORM



**Microsoft Power Platform**

The low code platform that spans Microsoft 365, Azure, Dynamics 365, and standalone apps.



**Power BI**  
Business analytics



**Power Apps**  
App development



**Power Automate**  
Process automation



**Power Virtual Agents**  
Intelligent virtual agents



**Power Pages**  
External-facing websites



Data  
connectors

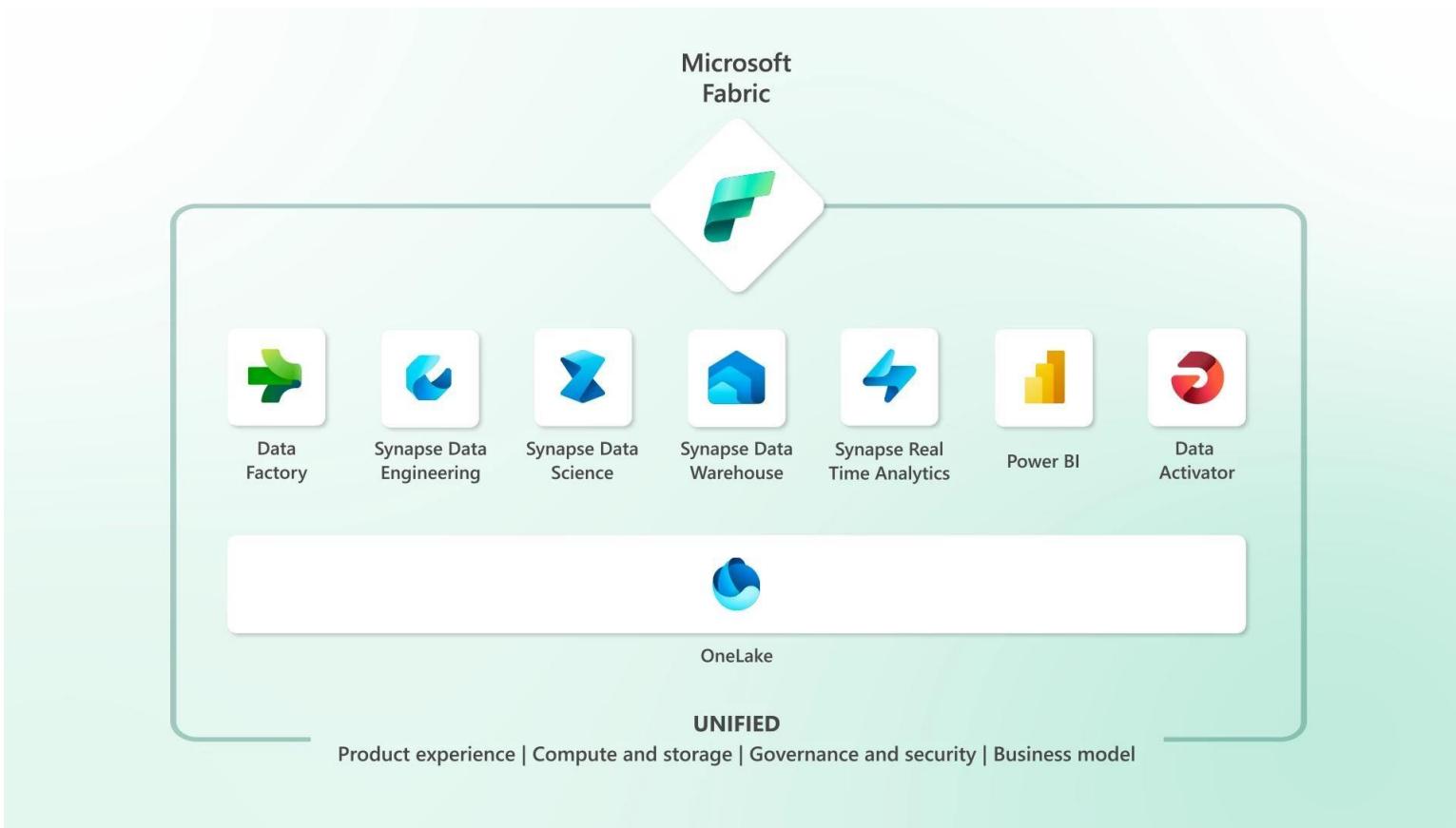


AI Builder



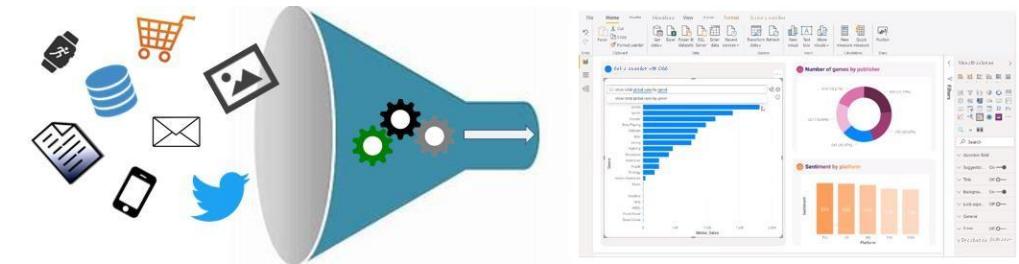
Dataverse

# MICROSOFT FABRIC



# CASOS DE USO

- Seguimiento del inventario
- Identificación de hábitos de compra
- Detección de tendencias y patrones de usuarios
- Recomendación de compras
- Definición de optimizaciones de precios
- Identificación y detención del fraude



# TIPOS DE ANÁLISIS

- Descriptivo
- Diagnóstico
- Predictivo
- Prescriptivo
- Cognitivo



The background image shows a man in a control room, likely a security or monitoring center, surrounded by multiple computer monitors displaying various feeds. He is seated at a desk with a keyboard and mouse. The room has a brick wall and several framed pictures or certificates hanging on it.

Preparación

Modelo

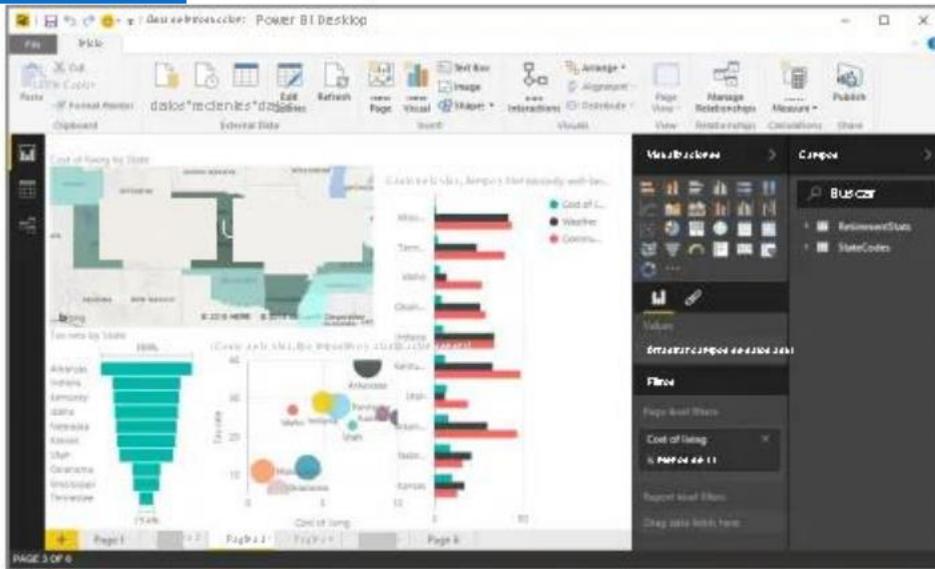
Visualización

Análisis

Administración

TAREAS

## Power BI Desktop



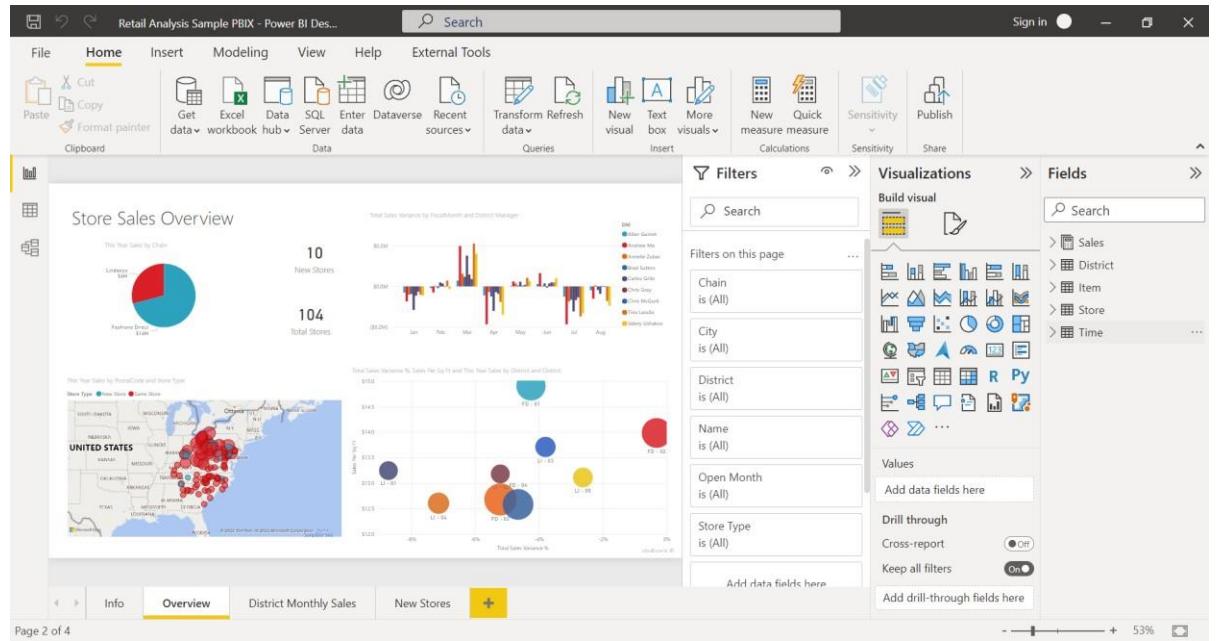
## Servicio Power BI



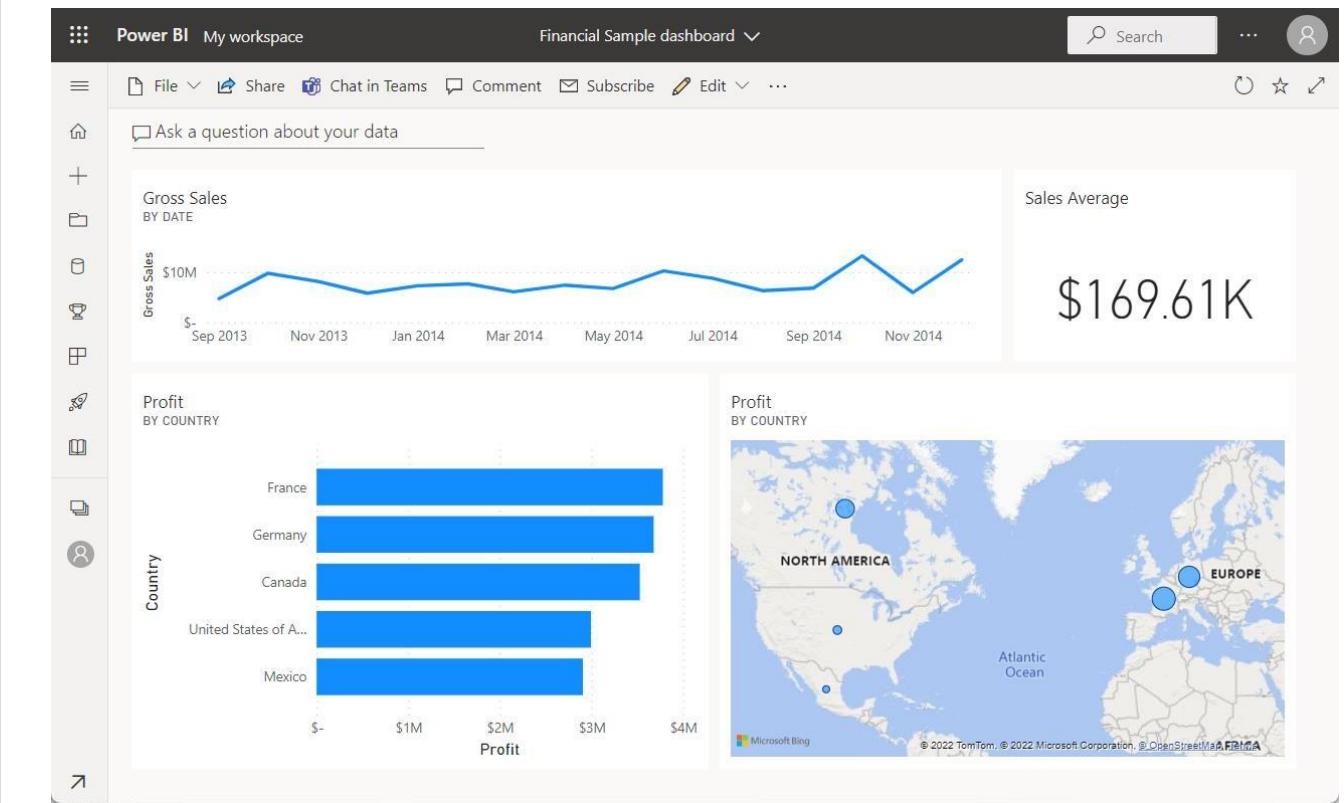
## Power BI Mobile

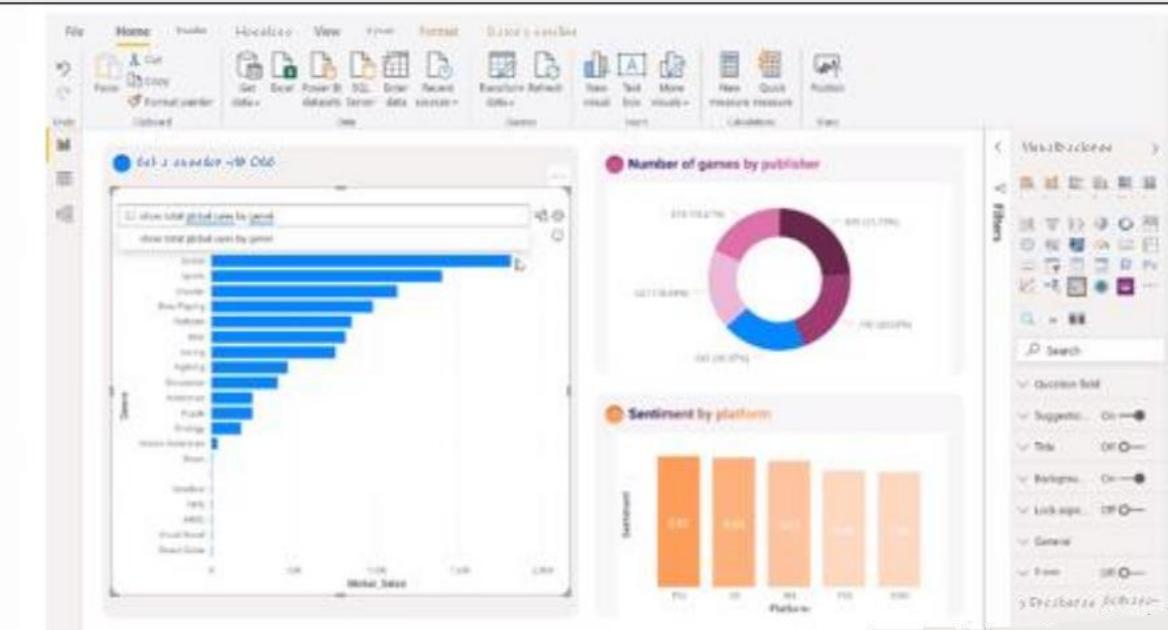


# POWER BI DESKTOP



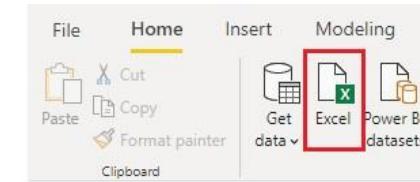
# POWER BI SERVICE





## FUENTES DE DATOS

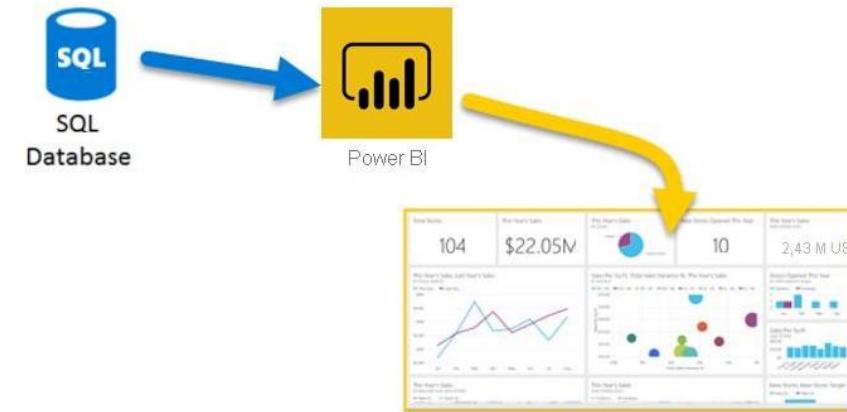
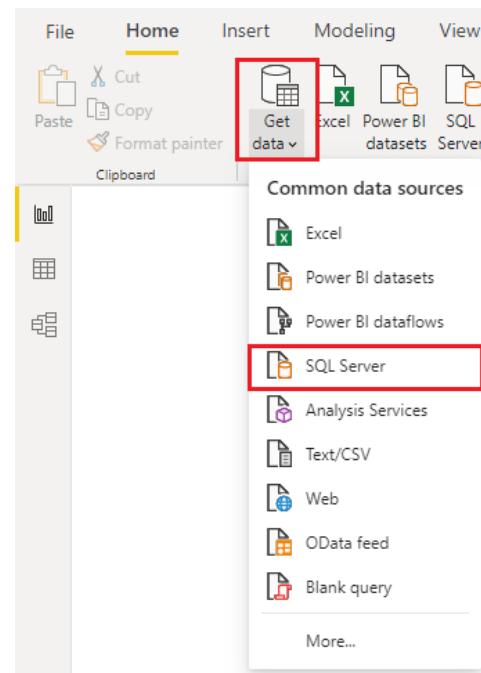
- Obtención de datos de archivo.
- Obtención de datos de orígenes de datos relacionales
- Obtención de datos de una base de datos NoSQL
- Obtención de datos de servicios en línea



## OBTENCIÓN DE DATOS DE ARCHIVO

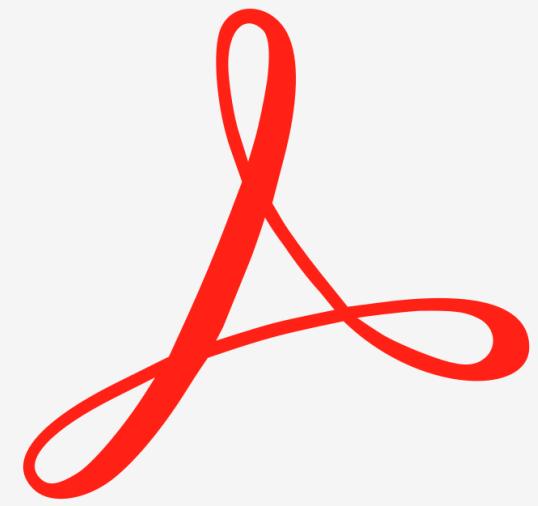
# CONEXIÓN A WEB





# SQL

PDF

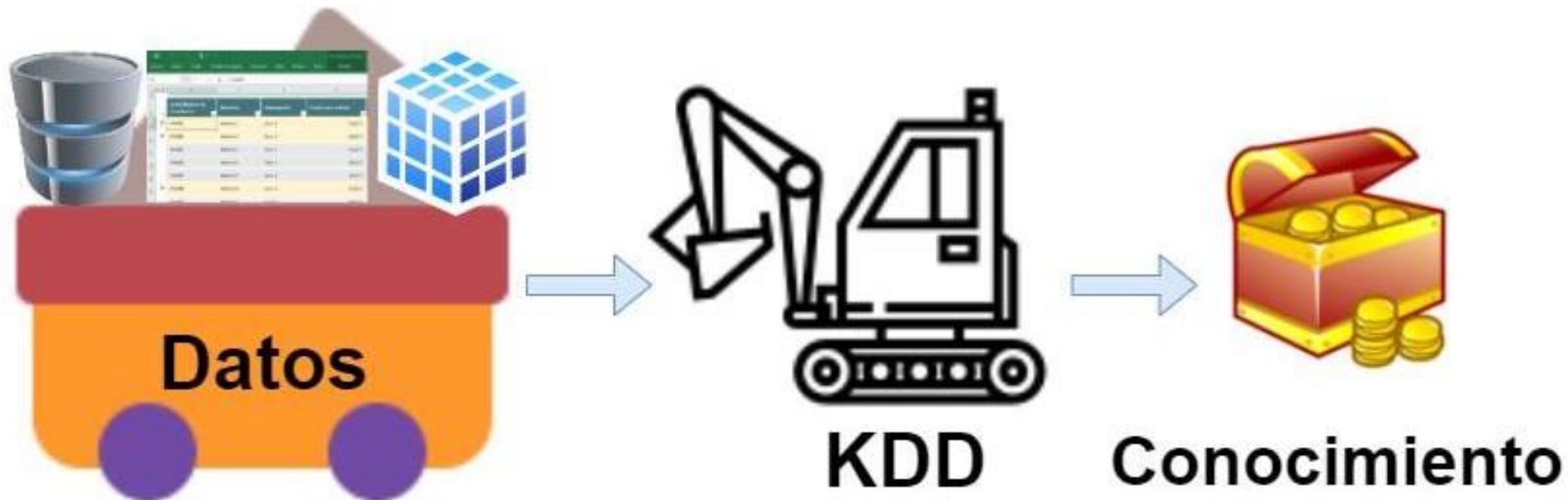


**PDF**

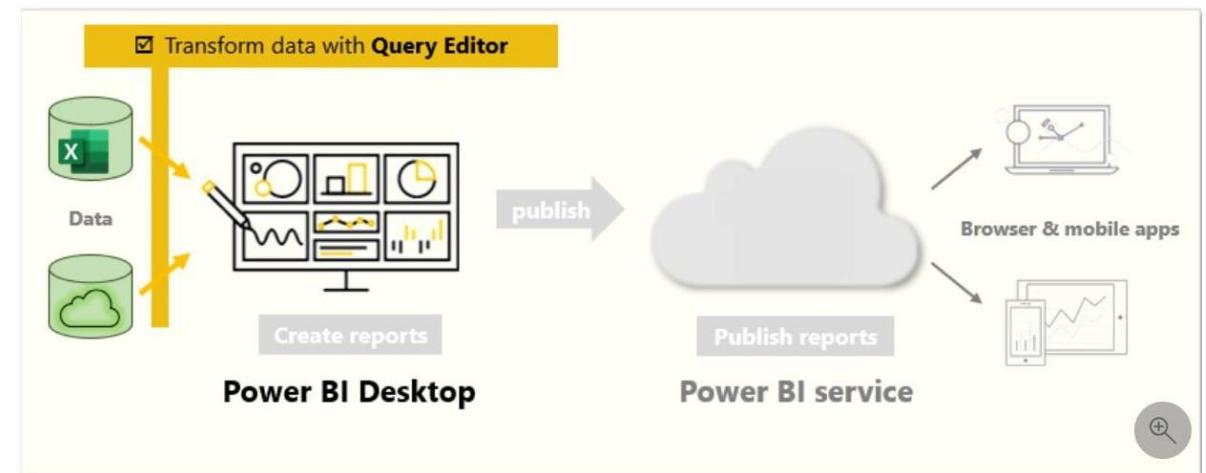
# EJERCICIO

- Consume los datos de esta web <https://www.worldometers.info/world-population/population-by-country/>
- Haz un gráfico de barras simple para ver los habitantes por país y comprobar que has ingestado correctamente los datos.

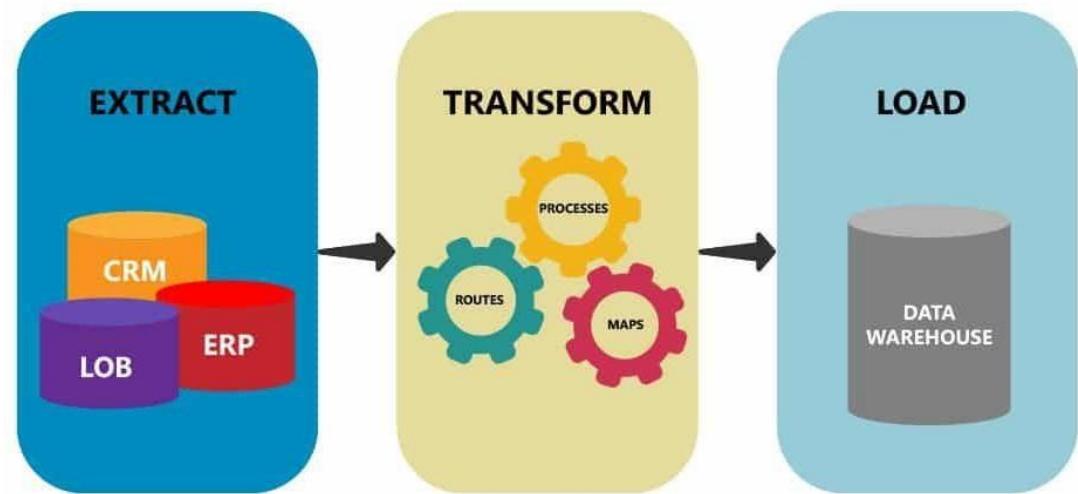
# PREPROCESAMIENTO



# POWER QUERY



# ETL



ETL - Extract, Transform, Load

Search

Display options ▾

 Invoices Order\_Details Order\_Details\_Extendeds Order\_Subtotals Orders Orders\_Qries Product\_Sales\_for\_1997 Products Products\_Above\_Average... Products\_by\_Categories Regions Sales\_by\_Categories Sales\_Totals\_by\_Amounts Shippers Summary\_of\_Sales\_by\_...

## Suppliers

 SupplierID	CompanyName	ContactName	ContactTitle	Address
1	Exotic Liquids	Charlotte Cooper	Purchasing Manager	49 Gilbert St.
2	New Orleans Cajun Delights	Shelley Burke	Order Administrator	P.O. Box 78934
3	Grandma Kelly's Homestead	Regina Murphy	Sales Representative	707 Oxford Rd.
4	Tokyo Traders	Yoshi Nagase	Marketing Manager	9-8 Sekimai Musashino-shi
5	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saave...	Export Administrator	Calle del Rosal 4
6	Mayumi's	Mayumi Ohno	Marketing Representative	92 Setsuko Chuo-ku
7	Pavlova, Ltd.	Ian Devling	Marketing Manager	74 Rose St. Moonie Ponds
8	Specialty Biscuits, Ltd.	Peter Wilson	Sales Representative	29 King's Way
9	PB Knäckebröd AB	Lars Peterson	Sales Agent	Kaloadagatan 13
10	Refrescos Americanas LTDA	Carlos Diaz	Marketing Manager	Av. das Americanas 12.890
11	Heli Süßwaren GmbH & Co. KG	Petra Winkler	Sales Manager	Tiergartenstraße 5
12	Plutzer Lebensmittelgroßmärkte AG	Martin Bein	International Marketing M...	Bogenallee 51
13	Nord-Ost-Fisch Handelsgesellschaft m...	Sven Petersen	Coordinator Foreign Mark...	Frahmredder 112a
14	Formaggi Fortini s.r.l.	Elio Rossi	Sales Representative	Viale Dante, 75
15	Norske Meierier	Beate Vileid	Marketing Manager	Hatlevagen 5
16	Bigfoot Breweries	Cheryl Saylor	Regional Account Rep.	3400 - 8th Avenue Suite 210
17	Svensk Sjöföda AB	Michael Björn	Sales Representative	Brovallavägen 231
	Ariastiques	Guylène Nodier	Sales Manager	203, Rue des Francs-Bourgeois
		Robb Merchant	Wholesale Account Agent	Order Processing Dept. 2100 Paul Re
		Ekaka	Owner	471 Serangoon Loop, Suite #402
			Sales Manager	Lyngbysild Fiskebakken 10

VENTAJAS

Cancel

Create

# EDITOR AVANZADO

## Sales Orders

Display Options ... ?

```
let
    Source = Sql.Database("localhost", "tsqlv4"),
    Sales_Orders = Source{[Schema="Sales",Item="Orders"]}[Data],
    #"Split Column by Delimiter" = Table.SplitColumn(Sales_Orders, "shipaddress", Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv), {"shipaddress.1", "shipaddress.2"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"shipaddress.1", type text}, {"shipaddress.2", type text}})
in
    #"Changed Type"
```

✓ No syntax errors have been detected.

1

form Add column View Help

Manage parameters Refresh Advanced editor Choose columns Remove columns Keep rows Remove rows Sort ABC 123 Data type: Text ▾ Use first row as headers ▾ Merge queries Append queries Combine files Map to entity CDM Insights

Parameters Query Manage columns Reduce rows Transform Combine CDM Insights

**Source{[Name = "Customers", Signature = "table"]}[Data]**

CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	null
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	null
ANTON	Antonio Moreno Taquería	Antonio Moreno		Mataderos 2312	México D.F.	null
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	null
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	null
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	null
BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	null
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid	null
BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	null
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen	BC
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London	null
CACTU	Cactus Comidas para llevar	Patricia Simpson	Sales Agent	Cerrito 333	Buenos Aires	null
CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.	null
CHINW	Chop-suey Chinese		Owner	Hauptstr. 29	Bern	null
FRANZ	Fränkische Feuerweissbierbrouwerij			Av. dos Lusíadas, 23	Sao Paulo	SP
GRUNF	Gremi Frutos do Mar			Berkeley Gardens 12 Brewery	London	null
ISLAT	Island Trading			21	Aachen	null
KUMAR	Kumaraguru Brothers			21	Nantes	null
LAZYK	Lazy K			Chenes		

Query settings

Name: Customers

Entity type: Custom

Applied steps:

- Source
- Navigation

Step

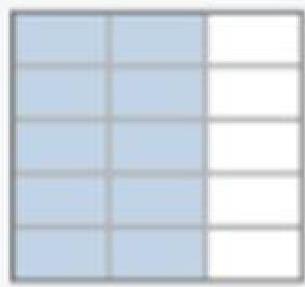
Cancel Save

# EDITOR DE POWER QUERY

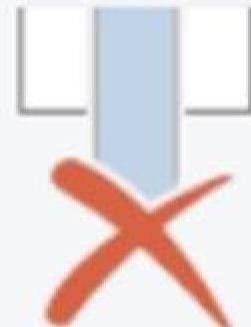
# ADMINISTRAR COLUMNAS



Power BI



Elegir  
columnas ▾    columnas ▾

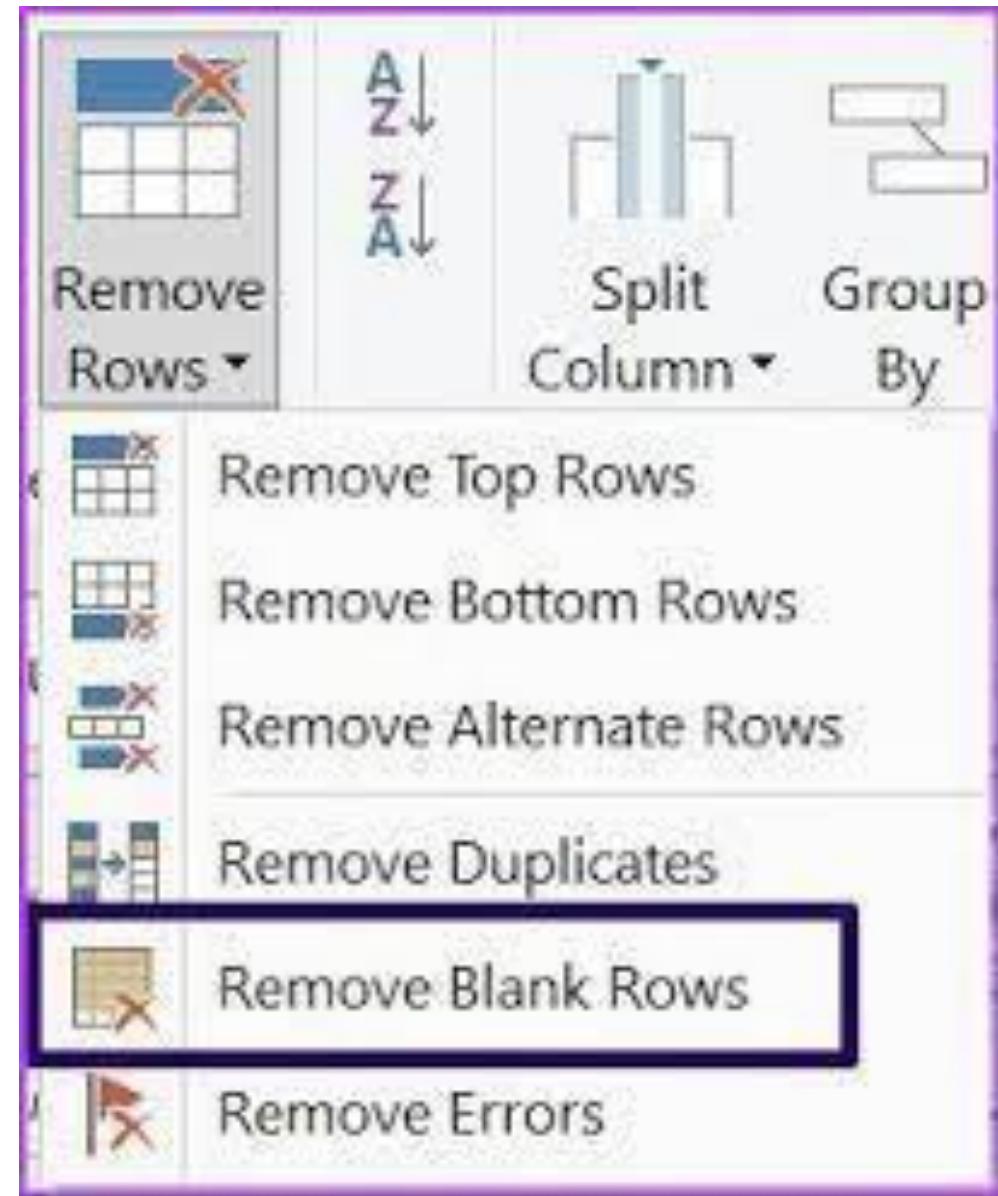


Quitar

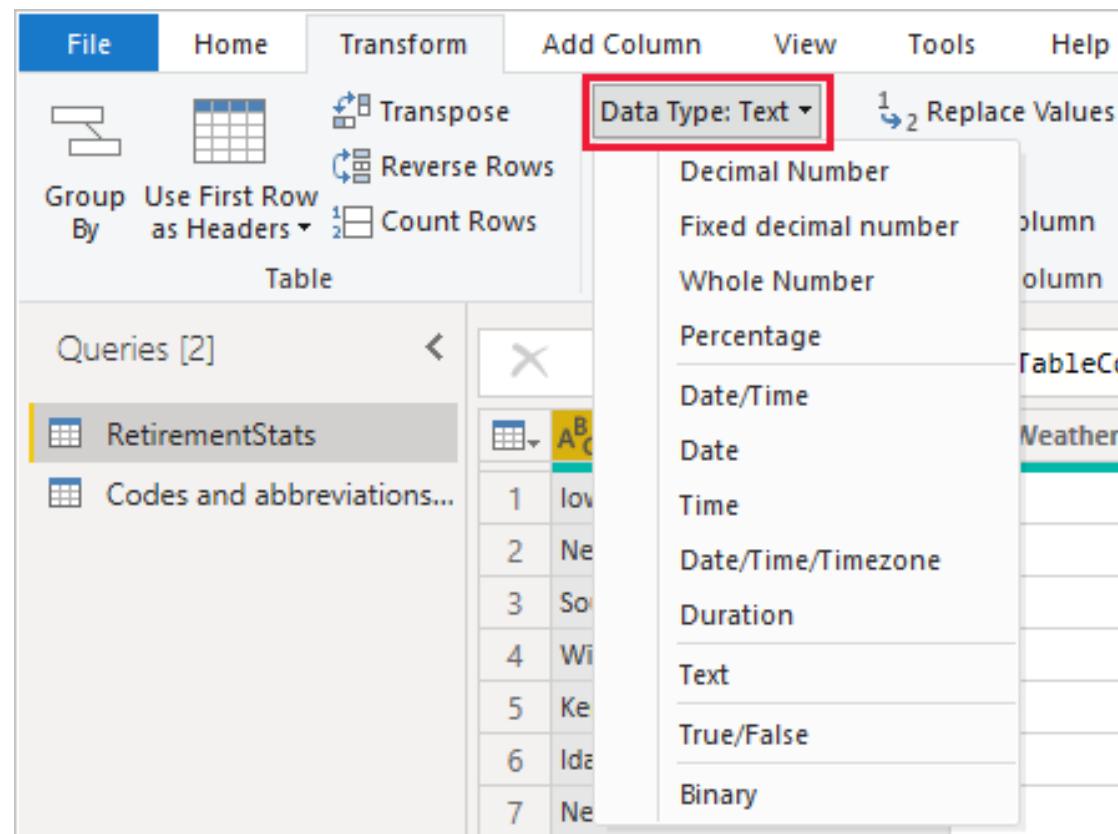
ELIMINAR COLUMNAS

columnas

# ELIMINAR FILAS



# CONVERTIR TIPO DE DATOS



# DIVIDIR COLUMNAS

The screenshot shows the Microsoft Power Query Editor interface. The top navigation bar includes File, Home, Transform (selected), Add Column, View, Tools, and Help. The Transform tab has several icons: Group By, Use First Row as Headers, Transpose, Reverse Rows, Detect Data Type, Fill, Rename, Unpivot Columns, Move, Pivot Column, Convert to List, Split Column, Format, Merge Columns, Extract, and Parse. A dropdown menu for 'Split Column' is open, listing options: By Delimiter, By Number of Characters, By Positions (which is highlighted with a red box), By Lowercase to Uppercase, By Uppercase to Lowercase, By Digit to Non-Digit, and By Non-Digit to Digit.

Table

Any Column

= Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText

Queries

	Column1
1	PA000420190504150
2	US94242019112487489
3	PA4144202001123758
4	US84292020031467895

# REEMPLAZAR VALORES

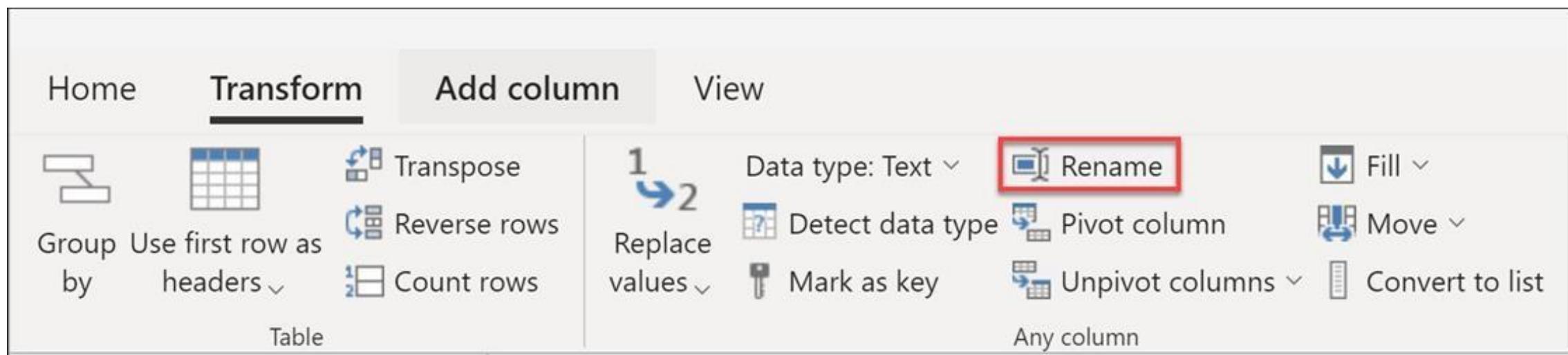
The screenshot shows the Power BI Data Editor interface. A context menu is open over a table containing four rows of data. The table has three columns: Account ID, Category Name, and Sales Goal. The data is as follows:

Account ID	Category Name	Sales Goal
1	120024 Category Name: Prime	
2	120025 Category Name: Main	
3	120026 Category Name: Base	
4	120027 Category Name: Lower	

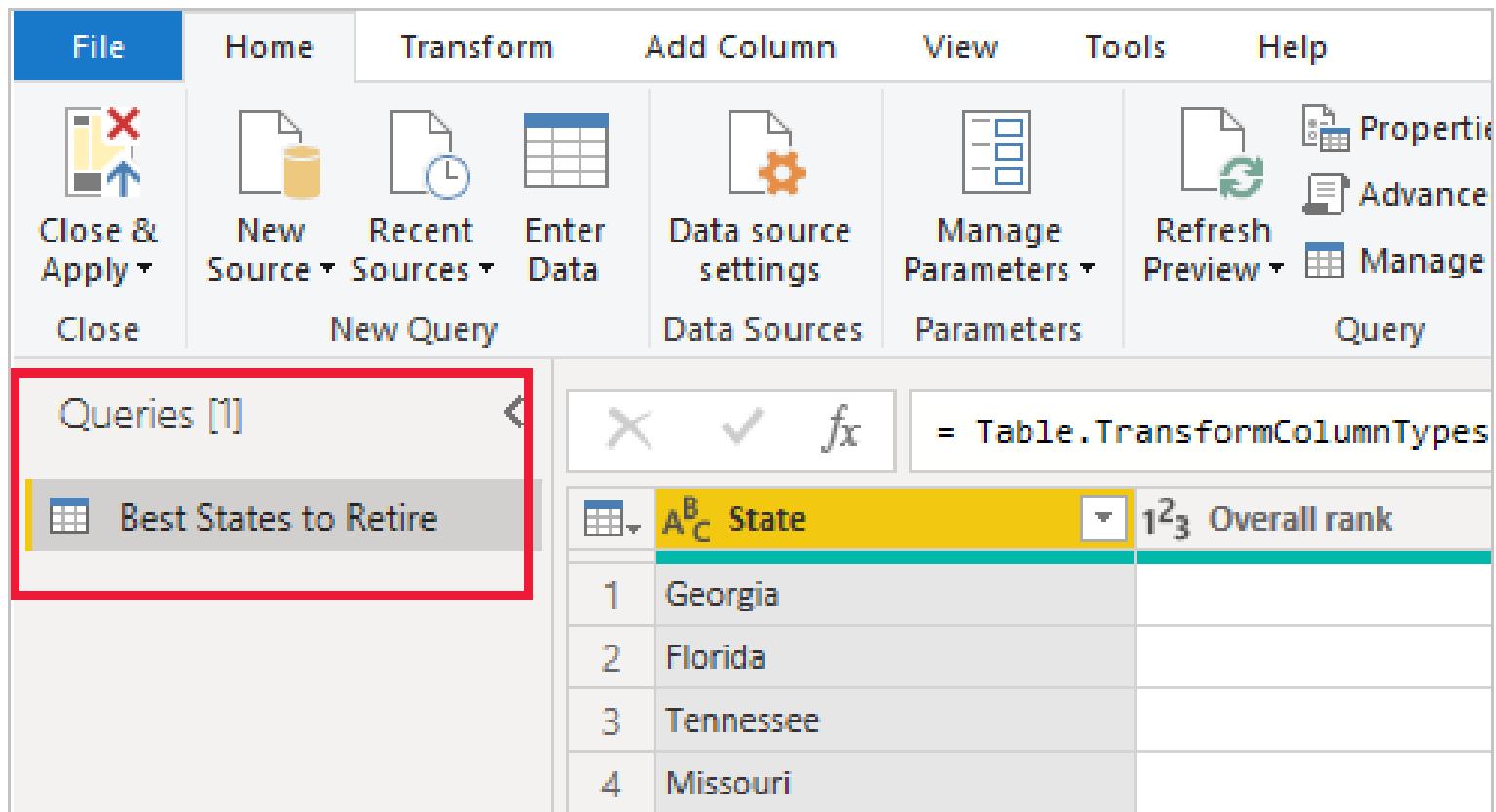
The context menu, located on the right side of the table, includes the following options:

- Copy
- Remove
- Remove Other Columns
- Duplicate Column
- Add Column From Examples...
- Remove Duplicates
- Remove Errors
- Change Type
- Transform
- Replace Values... (highlighted with a red box)
- Replace Errors...
- Group By...
- Fill
- Unpivot Columns
- Unpivot Other Columns
- Unpivot Only Selected Columns
- Rename...
- Move
- Drill Down
- Add as New Query

# CAMBIAR NOMBRE DE COLUMNAS



# APLICAR CAMBIOS



The screenshot shows the Power BI desktop interface. The ribbon at the top has tabs: File, Home, Transform, Add Column, View, Tools, and Help. Below the ribbon, there are several icons and their corresponding labels: Close & Apply (with Close and New Source dropdowns), Recent Sources (with New Query), Enter Data, Data source settings, Manage Parameters (with Refresh Preview dropdown), Properties (with Advance and Manage), and a Query button.

The main area displays a list of queries under the heading "Queries [1]". A single query, "Best States to Retire", is listed and highlighted with a red border. To the right of the query list is a preview pane showing a table with four rows. The columns are labeled "A", "B", "C", "State", "1", "2", "3", and "Overall rank". The data in the table is:

A	B	C	State	1	2	3	Overall rank
1			Georgia				
2			Florida				
3			Tennessee				
4			Missouri				

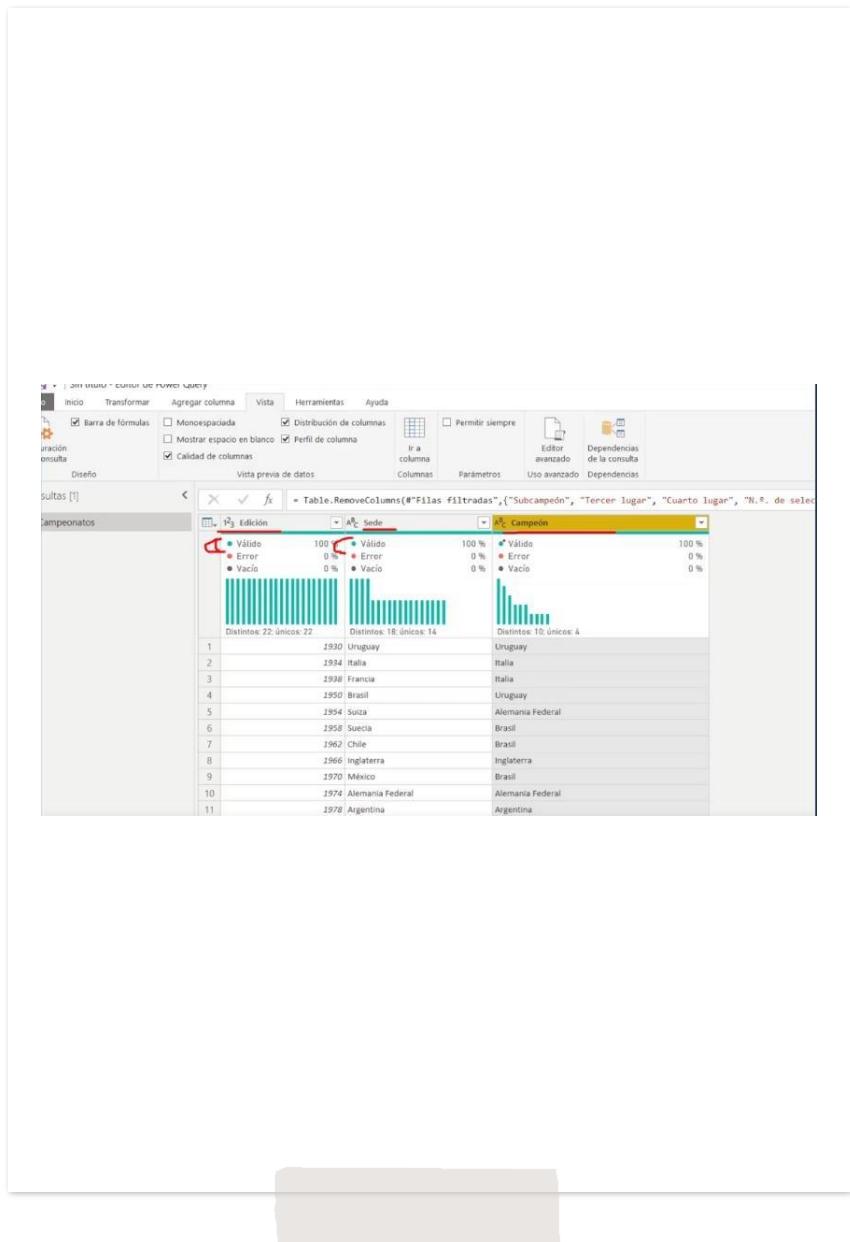
# EJERCICIO

Obtener información de esta url:  
[https://es.wikipedia.org/wiki/Copa\\_Mundial\\_de\\_F%C3%BAtbol](https://es.wikipedia.org/wiki/Copa_Mundial_de_F%C3%BAtbol) e importar la tabla de campeonatos. Generar una base de datos que permita identificar el equipo con mas campeonatos y la sede con mas mundiales realizados, debe de quedar como la adjunta

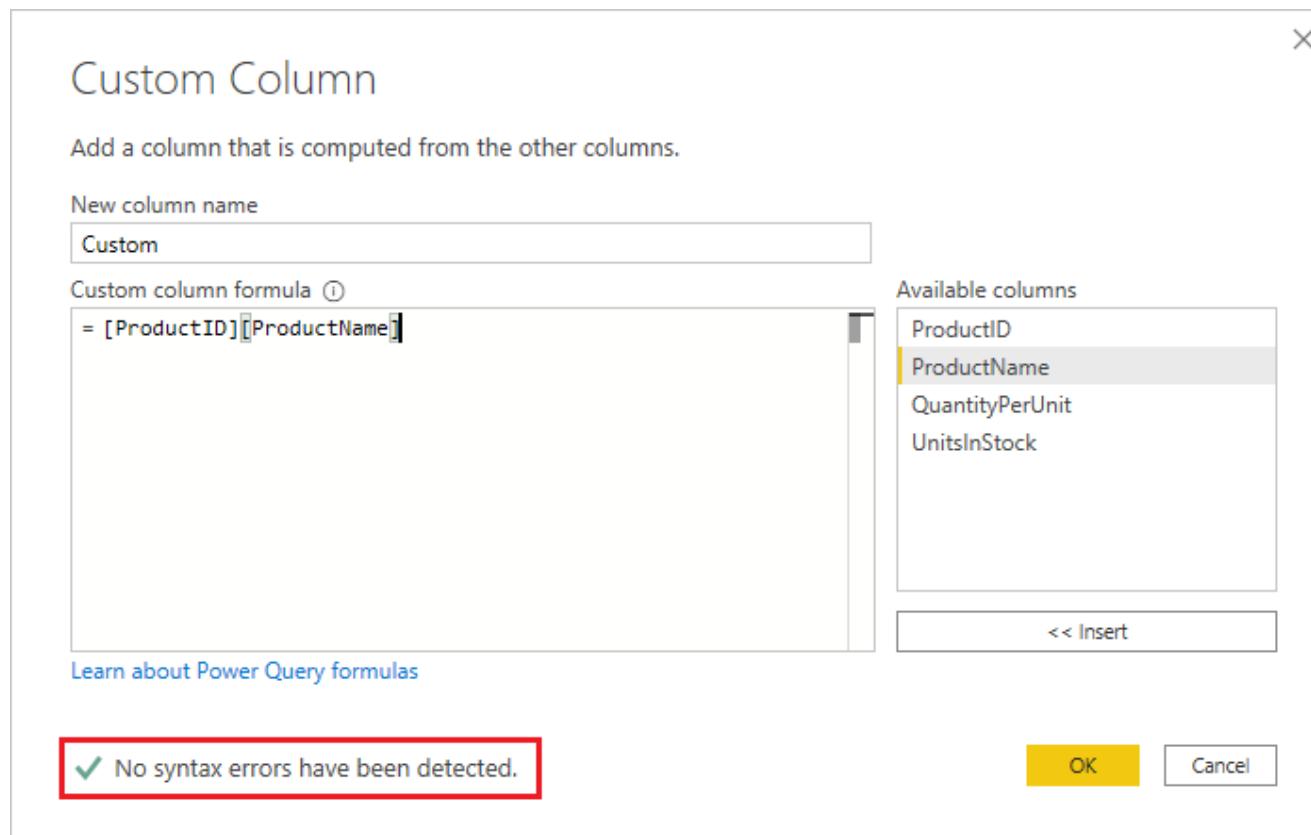
Los requisitos de la tabla son:

- La calidad de las columnas > 90%
- No deben haber columnas ni registros vacíos
- Eliminar columnas innecesarias
- Garantizar que los datos están de acuerdo a su tipo
- Reemplazar valores
- Cambiar nombre de columnas

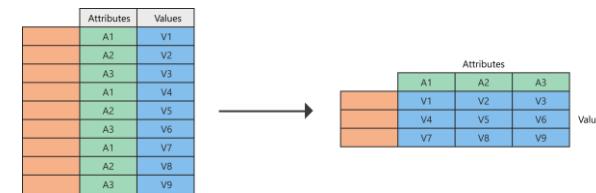
Como plus, podeis generar un grafico con cada una de esas informaciones



# COLUMNA PERSONALIZADA



	A <sub>B</sub> C	Country	Date	1 <sup>2</sup> <sub>3</sub>	Value
1	USA		6/1/2020		785
2	USA		7/1/2020		450
3	USA		8/1/2020		567
4	Canada		6/1/2020		357
5	Canada		7/1/2020		421
6	Canada		8/1/2020		254
7	Panama		6/1/2020		20
8	Panama		7/1/2020		40
9	Panama		8/1/2020		80



	A <sub>B</sub> C	Country	1 <sup>2</sup> <sub>3</sub>	6/1/2020	1 <sup>2</sup> <sub>3</sub>	7/1/2020	1 <sup>2</sup> <sub>3</sub>	8/1/2020
1	Canada					357	421	254
2	Panama					20	40	80
3	USA					785	450	567

# DINAMIZACION DE COLUMNAS

	A <sup>B</sup> <sub>C</sub> Country	1 <sup>2</sup> <sub>3</sub> 6/1/2020	1 <sup>2</sup> <sub>3</sub> 7/1/2020	1 <sup>2</sup> <sub>3</sub> 8/1/2020
1	USA	785	450	567
2	Canada	357	421	254
3	Panama	20	40	80

Attributes		
	A1	A2
	V1	V2
	V4	V5
	V7	V8
		V9



	Attributes	Values
1	A1	V1
2	A2	V2
3	A3	V3
4	A1	V4
5	A2	V5
6	A3	V6
7	A1	V7
8	A2	V8
9	A3	V9

	A <sup>B</sup> <sub>C</sub> Country	A <sup>B</sup> <sub>C</sub> Attribute	1 <sup>2</sup> <sub>3</sub> Value
1	USA	6/1/2020	785
2	USA	7/1/2020	450
3	USA	8/1/2020	567
4	Canada	6/1/2020	357
5	Canada	7/1/2020	421
6	Canada	8/1/2020	254
7	Panama	6/1/2020	20
8	Panama	7/1/2020	40
9	Panama	8/1/2020	80

# ANULAR DINAMIZACION DE COLUMNAS

# AGRUPAR DATOS

The screenshot shows the Power BI Advanced Editor interface. At the top, the ribbon toolbar includes options like Data source settings, Manage Parameters, Refresh Preview, Properties, Advanced Editor, and various data management tools such as Choose Columns, Remove Columns, Keep Rows, Remove Rows, Reduce Rows, Sort, Split Column, Group By, and Transform. A red circle highlights the 'Group By' button in the ribbon.

The main area displays a table with three rows:

	description	Table.Date	1.2 SUM_value
1	Amazon	1/1/2020	6
2	Amazon	1/2/2020	7
3	Morrisons	1/2/2020	15

A red arrow points from the 'Group By' button in the ribbon down to the 'Group By' dialog box.

**Group By**  
Specify the column(s) to group by and one or more outputs.  
 Basic  Advanced

description  
Table.Date  
Add grouping

New column name: SUM\_value  
Operation: Sum  
Column: Table.value  
Add aggregation

OK Cancel

# CONSOLIDAR CONSULTAS

Merge

Select tables and matching columns to create a merged table.

S&P 500 Component Stocks - List of...

Ticker symbol	Company	SEC filings	GICS Sector	GICS Sub Industry
MMM	3M Company	reports	Industrials	Industrial Conglomer
ABT	Abbott Laboratories	reports	Health Care	Health Care Equipment
ABBV	AbbVie	reports	Health Care	Pharmaceuticals
ANF	Abercrombie & Fitch Company A	reports	Consumer Discretionary	Apparel, Accessories
ACE	ACE Limited	reports	Financials	Property & Casualty

NYSE 2009 - daily

exchange	stock_symbol	date	stock_price_close	stock_gain_loss_dollar	stock_gain_loss_perce
NYSE	AA	39815	12.11	0	-0.0
NYSE	AA	39818	11.86	-0.25	-0.0
NYSE	AA	39819	12.12	0.26	0.0
NYSE	AA	39820	10.89	-1.23	-0.0
NYSE	AA	39821	11.36	0.47	0.0

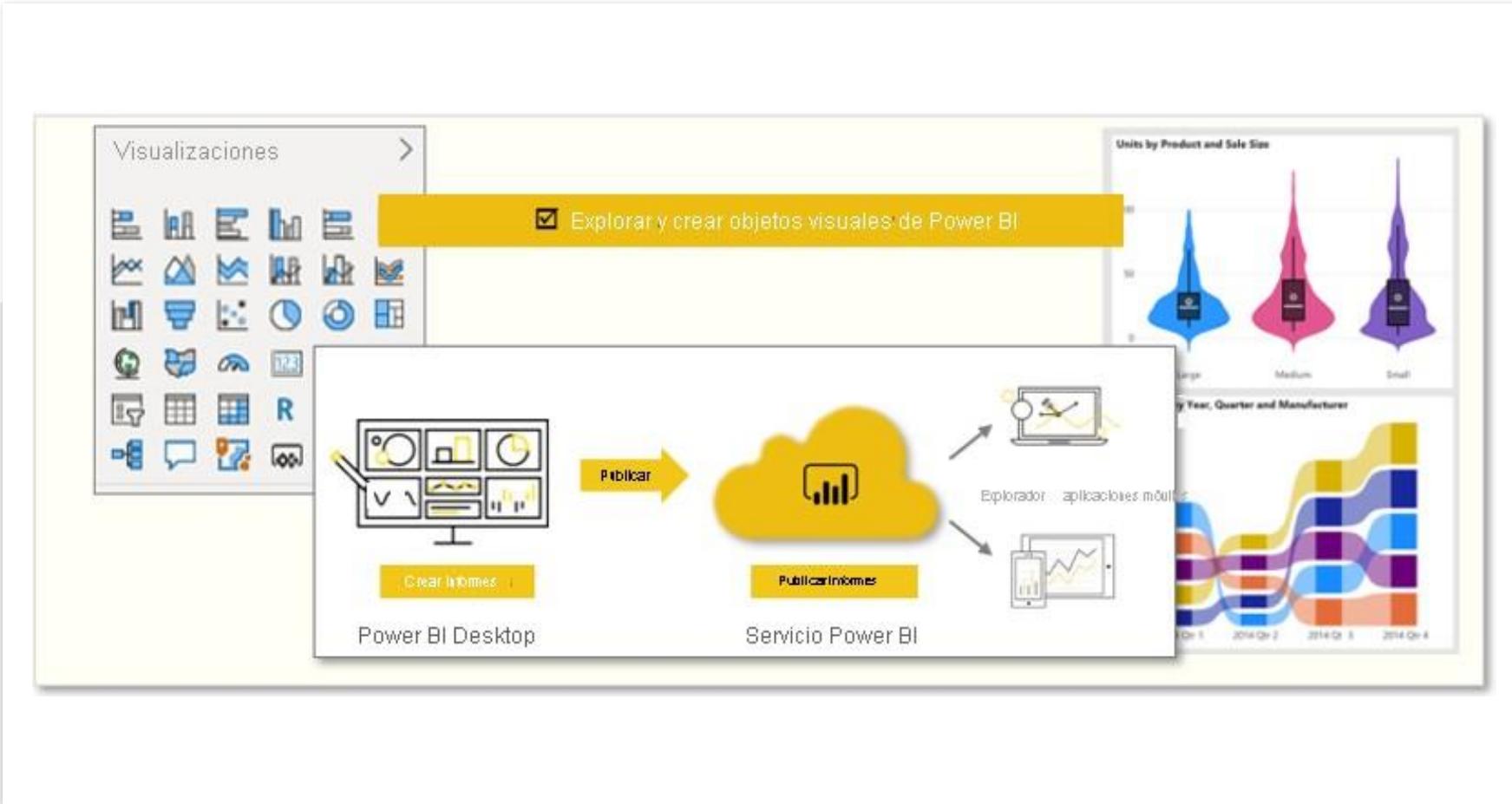
Only include matching rows

OK Cancel

# EJERCICIO

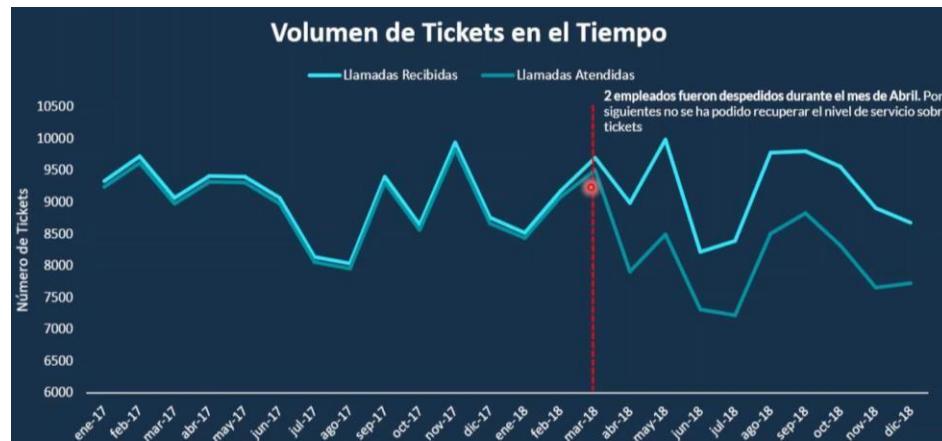
- Utilizando los datos de la carpeta POBLACION generar la integración y combinación de datos para contabilizar las personas por cada año y sexo
- Aplicar preprocessamiento, eliminar columnas y registros innecesarios, utilizar dinamización de columnas, y separa columnas

123 Año	A8C Sexo	1.2 Cantidad de Personas
● Válido ● Error ● Vacío	● Válido ● Error ● Vacío	● Válido ● Error ● Vacío
100 % 0 % 0 %	100 % 0 % 0 %	100 % 0 % 0 %
Distintos: 4; únicos: 0	Distintos: 2; únicos: 0	Distintos: 8; únicos: 8
1	2000 Hombres	47592253
2	2000 Mujeres	49891159
3	2005 Hombres	50249955
4	2005 Mujeres	53013433
5	2010 Hombres	54855231
6	2010 Mujeres	57481307
7	2020 Hombres	61473390
8	2020 Mujeres	64540634

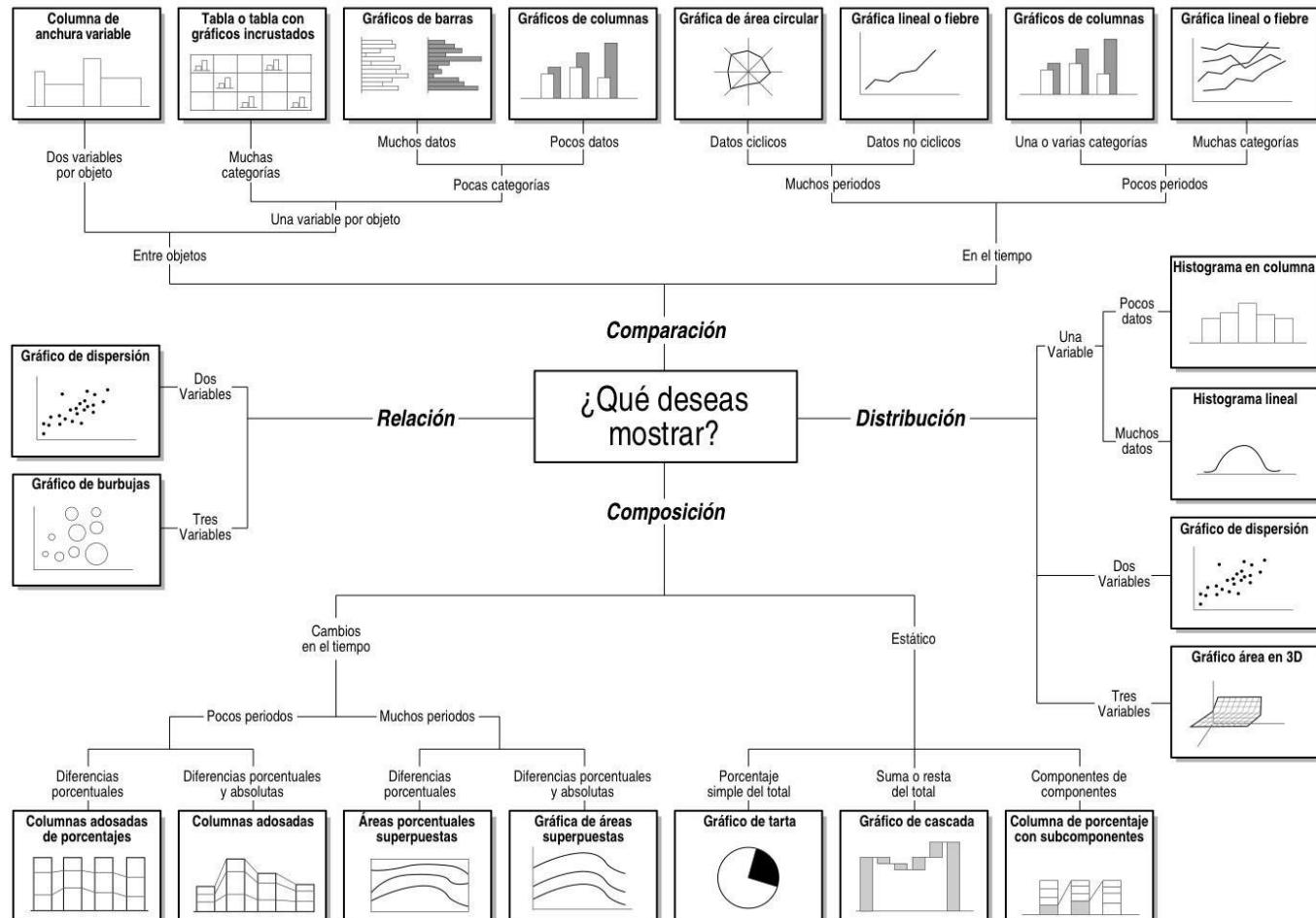


# VISUALIZACIONES

# ELECCIÓN DEL BUEN GRÁFICO



# ¿Qué gráfico elegir?



# CAMPOS Y FORMATO

The screenshot shows the Power BI interface with two main panes. On the left is the 'Fields' pane, which lists various data sources and their fields. On the right is the 'Visualizations' pane, which contains a grid of icons representing different chart types. A red box highlights the 'Axis' icon in the Visualizations pane, and another red box highlights the 'Month' field in the Fields pane.

Visualizations

Fields

Search

SalesFact

Sentiment

Date

Geo

Manufacturer

Manufacturer

MfgisVanArsdel

Product

Category

Count of Prod...

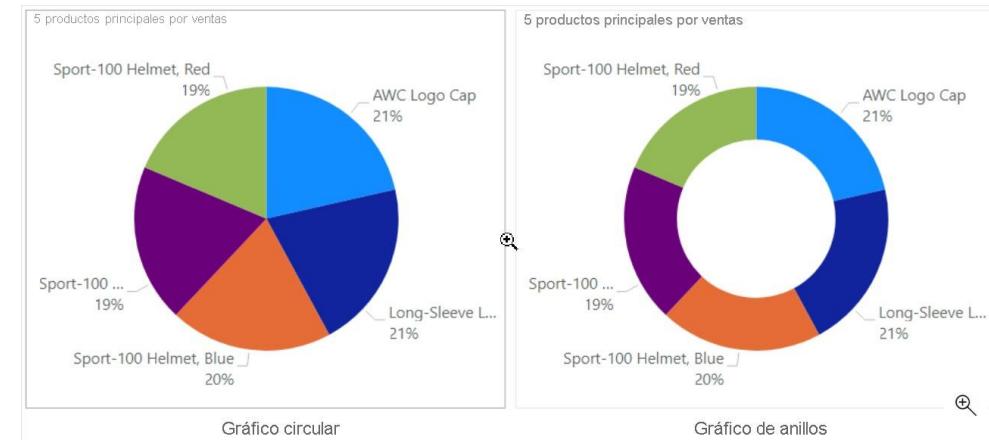
Filters

Axis

Month

Legend

Manufacturer



País Importe de ventas

País	Importe de ventas
Estados Unidos	\$7,390,464
France	\$773,445
Alemania	\$493,628
<b>Total</b>	<b>\$8,550,077</b>

Visualizaciones Campos

Filtros

Búsqueda

Territorio

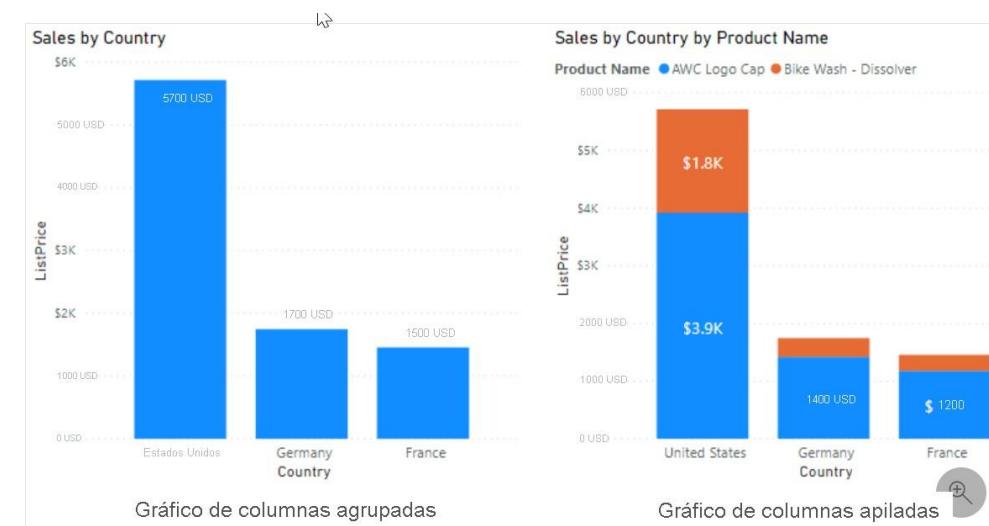
- Ciudad
- Ciudad (grupos)
- Country
- Europe
- Group
- Norte America
- Other

Valores

País

Importe de ventas

Importe...

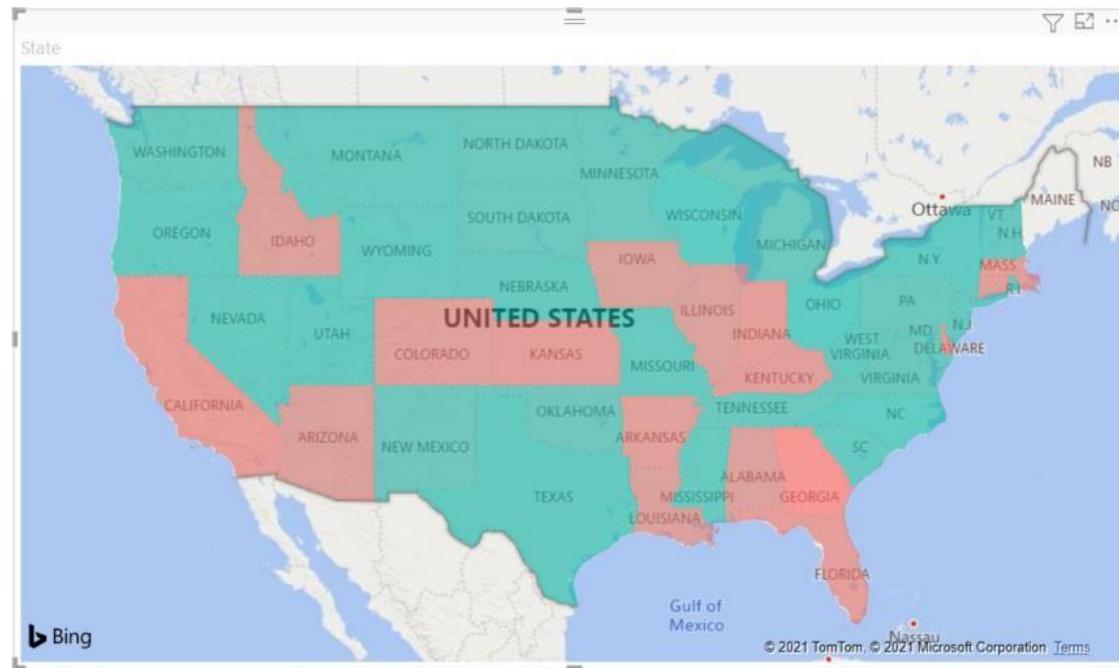




# EJERCICIO

Usando la base de datos de FIFA 23, (extraída de Kaggle) en formato CSV genera las siguientes visualizaciones:

- Grafico de columnas que nos permita identificar las nacionalidades mas frecuentes de los jugadores
- Grafico de barras que permita identificar que clubes aportan mas jugadores al listado
- Grafico que identifique la cantidad de jugadores que se han unido al juego año a año, usa la columna [Joined On]



# USO DE MAPAS



## EJERCICIO

- Utilizar el siguiente enlace para obtener información [https://en.wikipedia.org/wiki/List\\_of\\_busiest\\_airports\\_by\\_passenger\\_traffic](https://en.wikipedia.org/wiki/List_of_busiest_airports_by_passenger_traffic) acerca de los aeropuertos mas concurridos en el mundo.
- Generar un mapa de burbujas que nos ayude a identificar donde se encuentran los aeropuertos mas concurridos
- Limpiar los datos correctamente e identificar el tipo de columna para la ubicacion

Date

Año

Trimestre

Month

Día

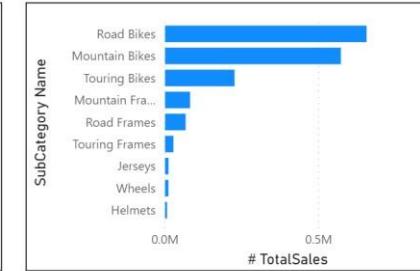
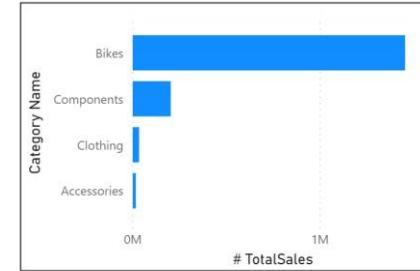
✓ ✗

✗

✗

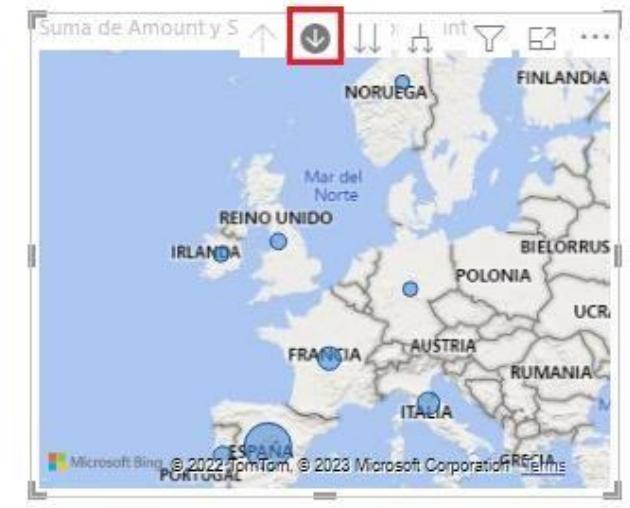
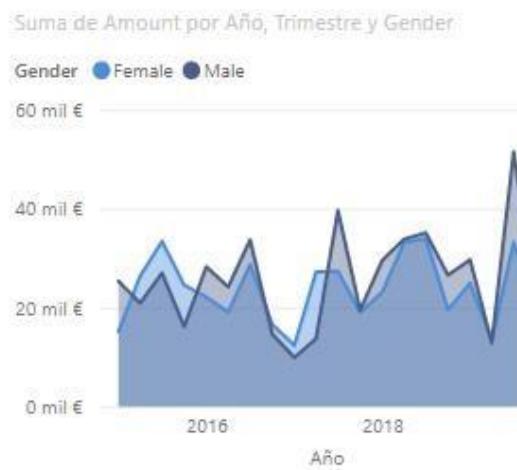
✗

✗



# JERARQUIAS

# NAVEGAR EN PROFUNDIDAD



The screenshot illustrates the process of adding a custom visual to Power BI. On the left, the 'Visualizations' pane shows various built-in visual types like Bar, Line, and Map. A red arrow points from the 'Get more visuals' button in the 'More options' menu to the 'AppSource' page on the right. The 'AppSource' page displays the 'Bullet Chart' app by Microsoft Corporation, which is PBI Certified and has a rating of 3.6 (36). The 'Add' button is highlighted with a red box, indicating the next step in the workflow.

Visualizations > Fields

Build visual

Get more visuals

Import a visual from a file

Remove a visual

Restore default visuals

AppSource | Apps for Power BI visuals

< Apps

Bullet Chart

Microsoft Corporation

3.6 (36)

PBI Certified

Overview Ratings + reviews

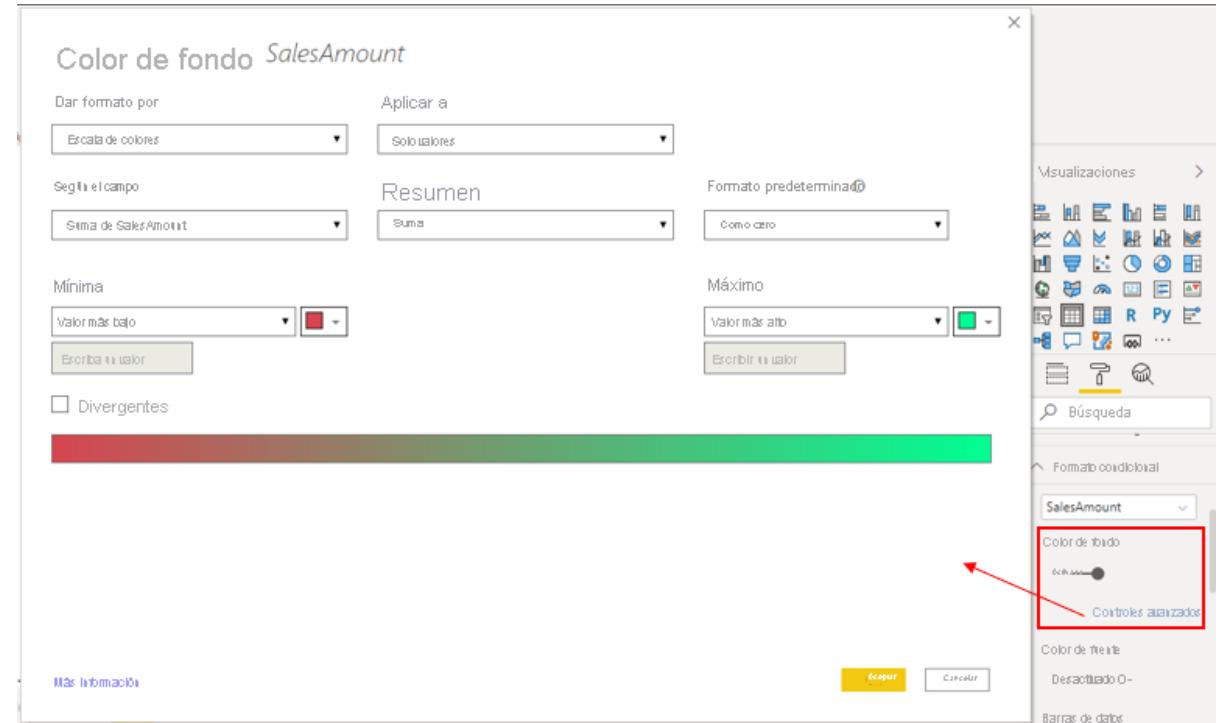
A bar chart with extra visual elements

# PERSONALIZADO

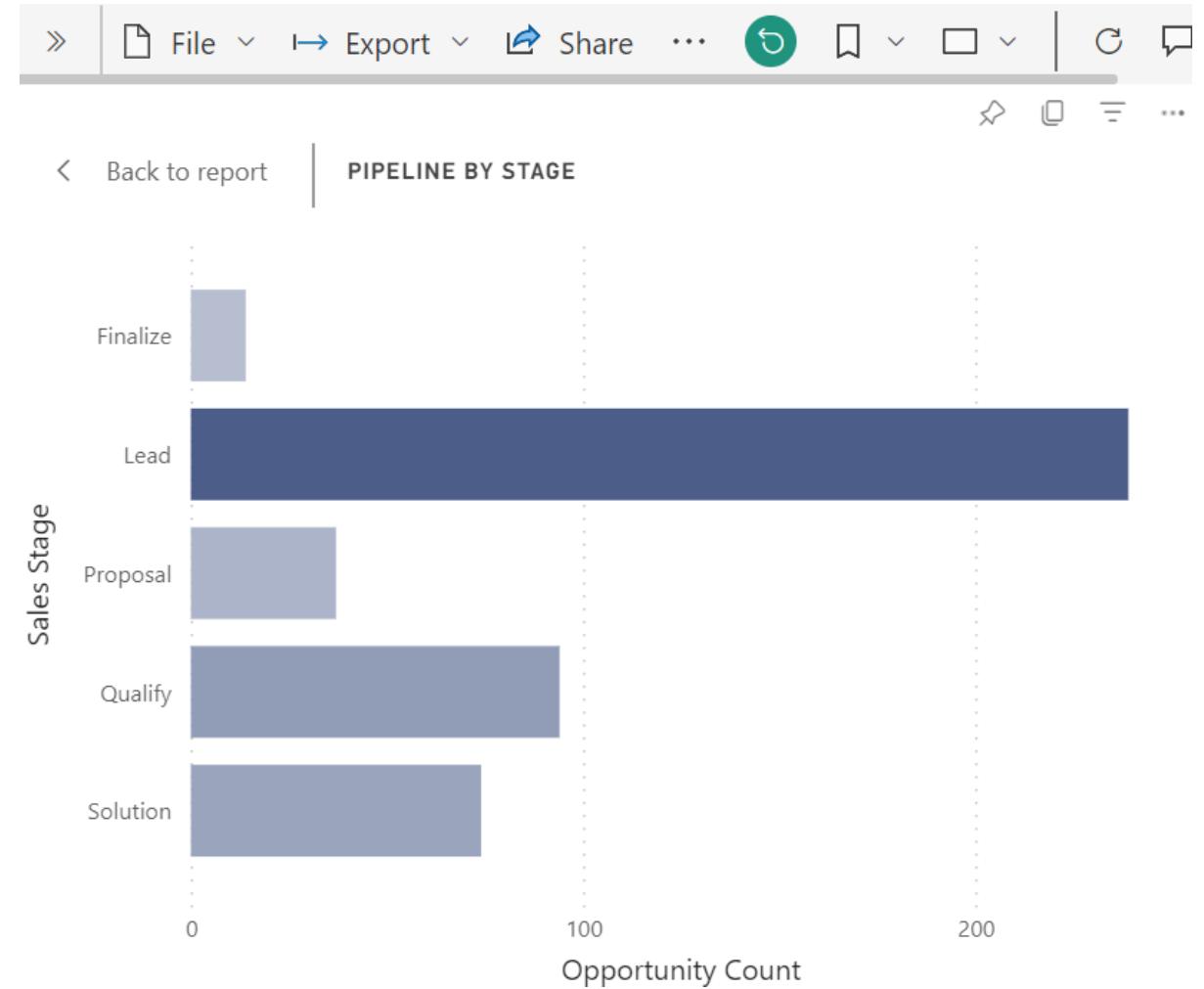


# R & PYTHON

# FORMATO CONDICIONAL



# ORDENAR GRAFICOS

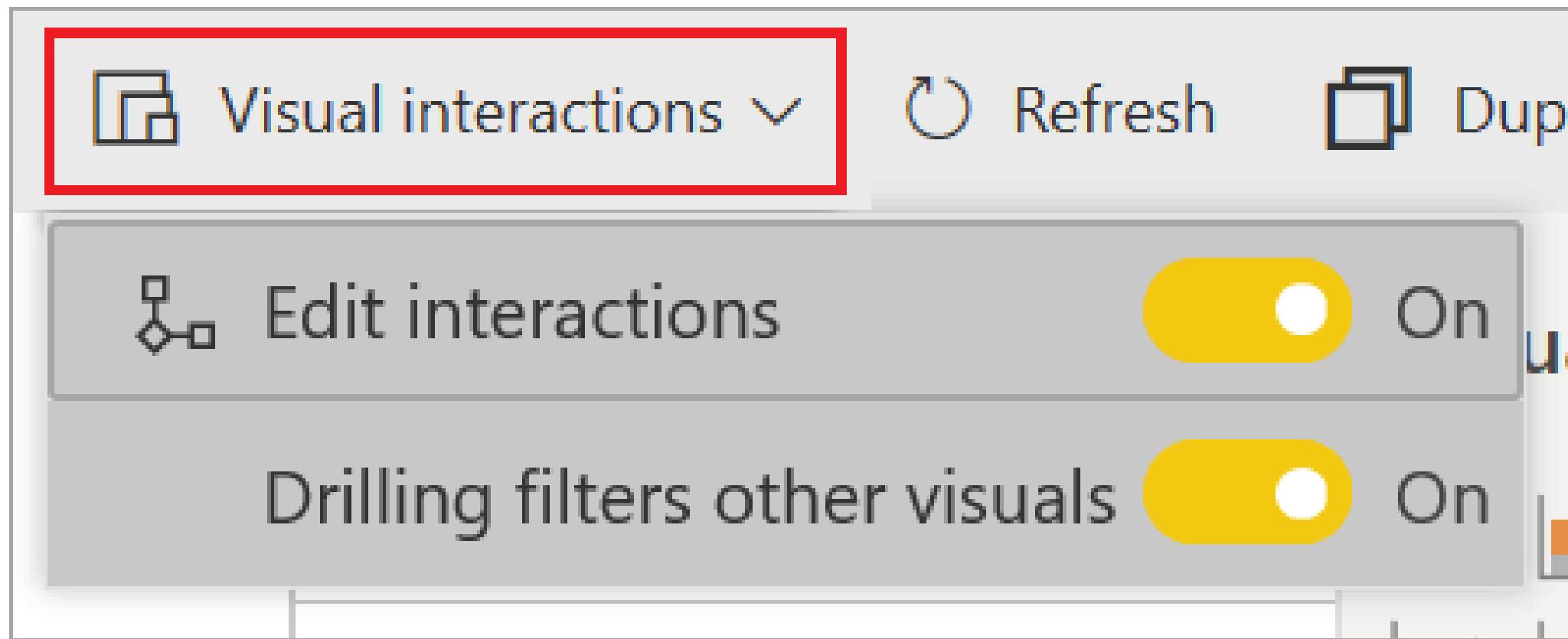


# SEGMENTACIÓN

The screenshot shows a user interface for segmenting data across four categories: Year, Category, Region, and Age.

- Año:** A horizontal navigation bar with tabs for 2016, 2017, 2018, and 2019. The 2018 tab is selected and highlighted in dark gray.
- Categoría:** A list of product categories with corresponding color-coded squares:
  - Accessories (light blue)
  - Bikes (dark blue)
  - Clothing (light green)
  - Components (light orange)
- Comarca:** A dropdown menu set to "Todas".
- Edad:** An age range selector with two input fields containing "33" and "103", connected by a horizontal slider.

# INTERACCIÓN ENTRE VISUALIZACIONES



# EJERCICIO

Utilizad la BBDD de SuperStore  
Dataset.xlsx. Generar visualizaciones de  
filtro cruzado entre los distintos  
elementos

- ¿Cuál es la cantidad de ventas de “Furniture” para el estado de New York?
- ¿Cuál es el producto de mayor demanda en el segmento de “Home Office”?
- ¿En que segmento y estado se vende con mayor frecuencia el segmento de Corporate?



# FILTROS

The screenshot shows the 'Filters' pane in Power BI, which is part of the 'Visualizations' tab. The pane displays three main sections: 'Filtros en este objeto visual' (Filters in this visual), 'Filtros de esta página' (Filters on this page), and a section for adding data fields.

**Filtros en este objeto visual:**

- Region es (Todo)
- SalesAmount is (All)

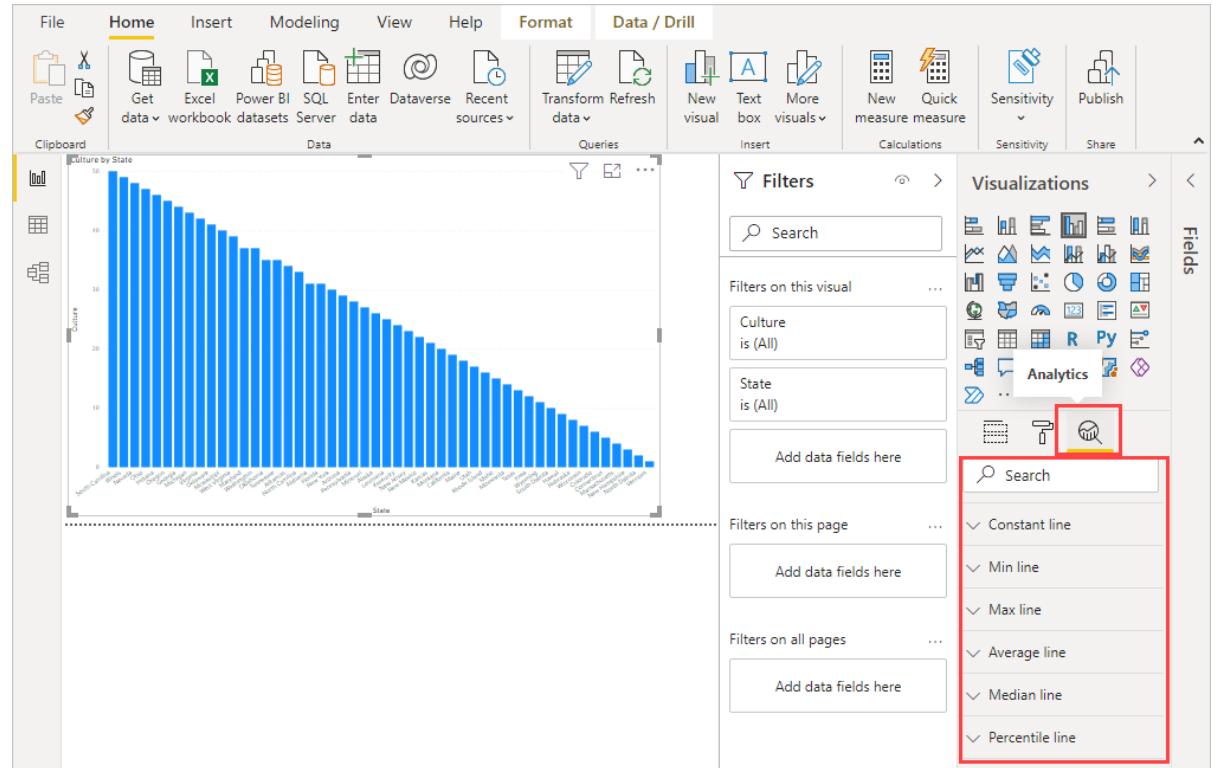
**Filtros de esta página:**

- DueDate - Year es (Todo)
  - Tipo de filtro: Filtrado avanzado
  - Mostrar elementos cuando el valor: es menor que

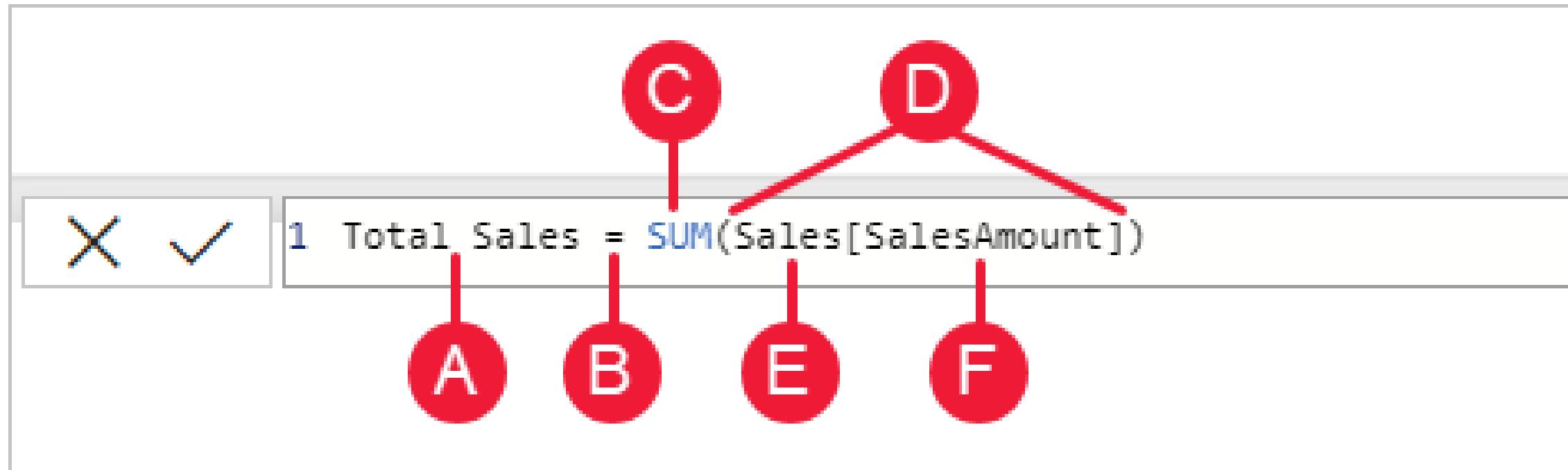
**Visualizaciones:** Shows various visualization icons.

**Campos:** Shows a list of fields from the 'SalesDataByRegion' table, with 'Year' selected. Other fields listed include DueDate, Date Hier., Year, Quarter, Ous Mes, Day, EmployeeKey, OrderDate, OrderQuan..., ProductKey, PromotionK..., Region, ResellerKey, SalesAmount, SalesOrderL..., SalesOrder..., and SalesTerrito... .

# ANALÍTICA



# DAX



# REFERENCIA FUNCIONES

## Agregación:

SUM  
PROMEDIO  
MIN  
MAX  
DIVIDE

## Recuento:

COUNT  
COUNTA  
COUNTBLANK  
COUNTRROWS  
DISTINCTCOUNT

## Lógicas:

Y  
O  
NOT  
IF  
IFERROR

## Información:

ISBLANK  
ISNUMBER  
ISTEXT  
ISNONTEXT  
ISERROR

## Texto:

CONCATENATE  
REPLACE  
SEARCH  
UPPER  
FIXED

## Fecha:

DATE  
HOUR  
NOW  
EOMONTH  
WEEKDAY

# MEDIDA RAPIDA

The screenshot shows the Power BI interface with three main panes:

- Filtros (Filters):** A list of filters applied to the page, including:
  - Chain (Todos)
  - City (Ciudad) (Todos)
  - District (Todos)
  - Nombre (Todos)
  - Open Month (Todos)
  - Store Type (Todos)
- Visualizaciones (Visualizations):** A grid of visualization icons.
- Campos (Fields):** A pane showing fields from various tables:
  - Búsqueda (Search):
  - Ventas (Sales):
  - District (District):
  - Elemento (Element):
  - Store (Store):
    - Average Selling Price (highlighted with a pink box)

A context menu is open for the 'Average Selling Price' field, with the 'Nueva medida rápida' (Quick Measure) option highlighted with a pink box. Other options in the menu include:

- Activar (Activate)
- Nueva medida (New Measure)
- Nueva columna (New Column)
- Nueva medida rápida (Quick Measure) (highlighted)
- Cambiar nombre (Change Name)
- Eliminar (Delete)
- Ocultar (Hide)
- Ver ocultos (View Hidden)
- Mostrar todo (Show All)
- Contraer todo (Collapse All)
- Expandir todo (Expand All)
- Agregar a filtros (Add to Filters)
- Agregar a la obtención de detalles (Add to Details Acquisition)
- PostalCode (Postal Code)

# COLUMNAS CALCULADAS

The screenshot shows a data grid interface with a toolbar at the top featuring tabs for 'Estructura' (Structure), 'Formato' (Format), 'Propiedades' (Properties), and 'Ordenar' (Sort). A search bar labeled 'Columna =' is positioned above the grid. The grid displays six rows of data with four columns each. The columns are represented by icons: a blue square for the first column, a red square for the second, a green square for the third, and a yellow square for the fourth. The data is as follows:

10248	11	12	\$14
10248	42	10	\$9.8
10248	72	5	\$34.8
10249	14	9	\$18.6
10249	51	40	\$42.4
10250	41	10	\$7.7

# EJERCICIO

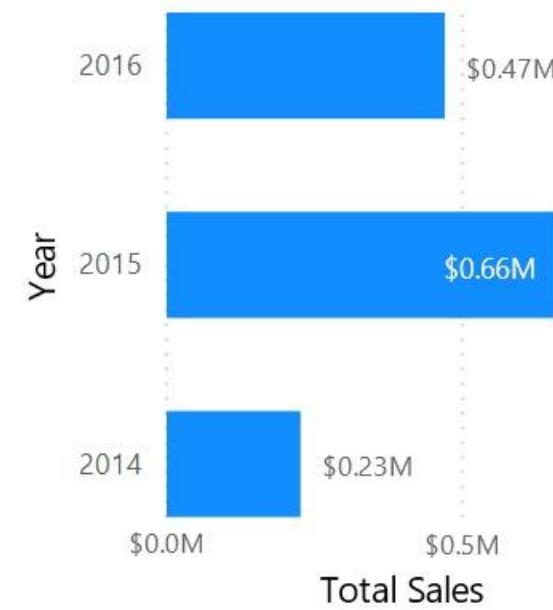


# CONTEXTO

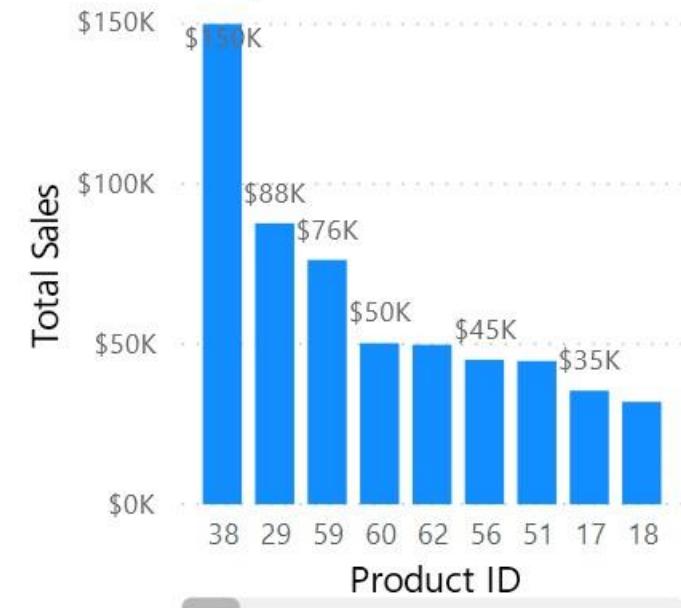
Total Sales



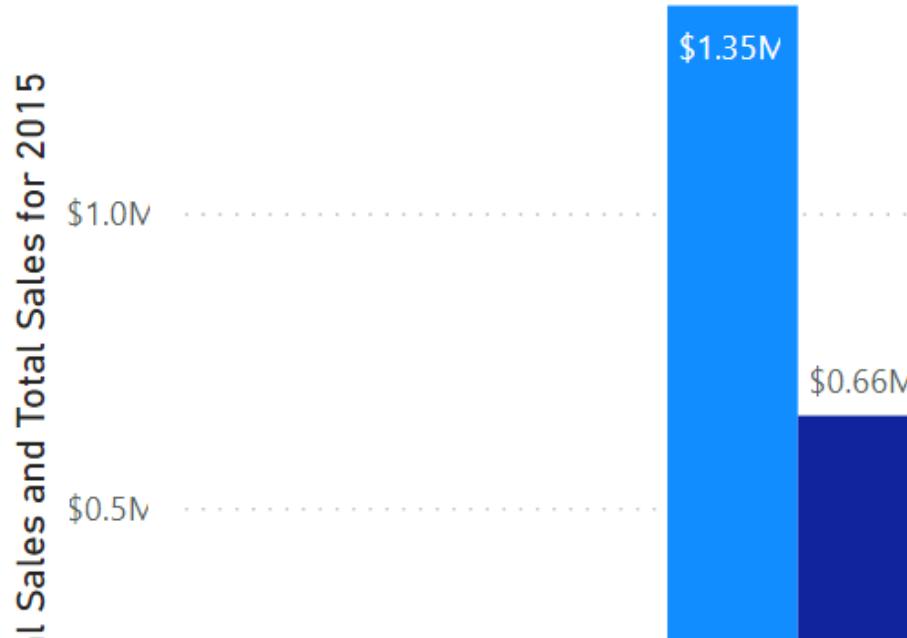
Total Sales by Year



Total Sales by Product ID



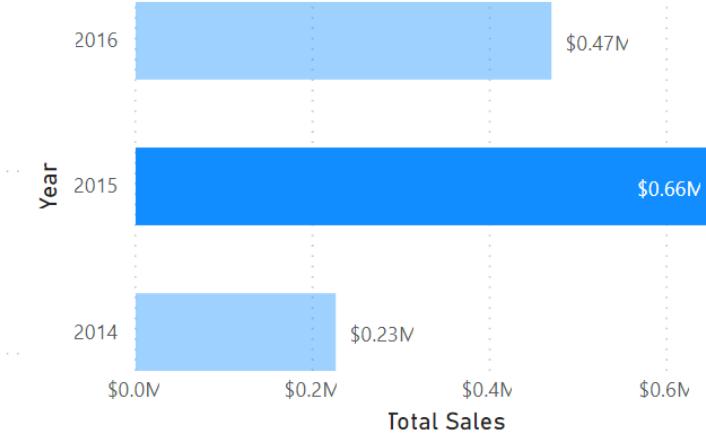
Total Sales Total Sales for 2015



and Total Sales for 2015

Total Sales for 2015

Total Sales by Year

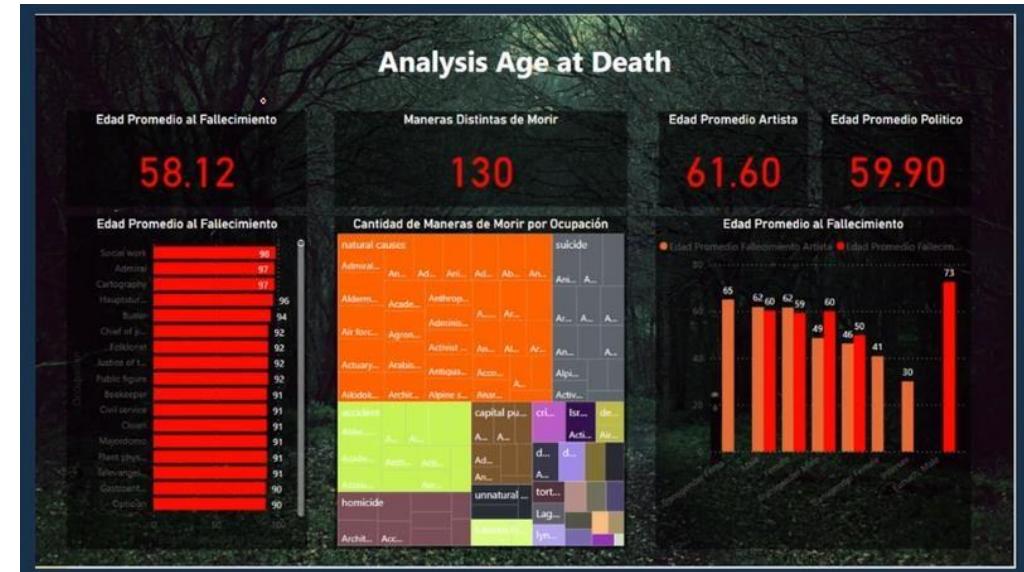


# CALCULATE

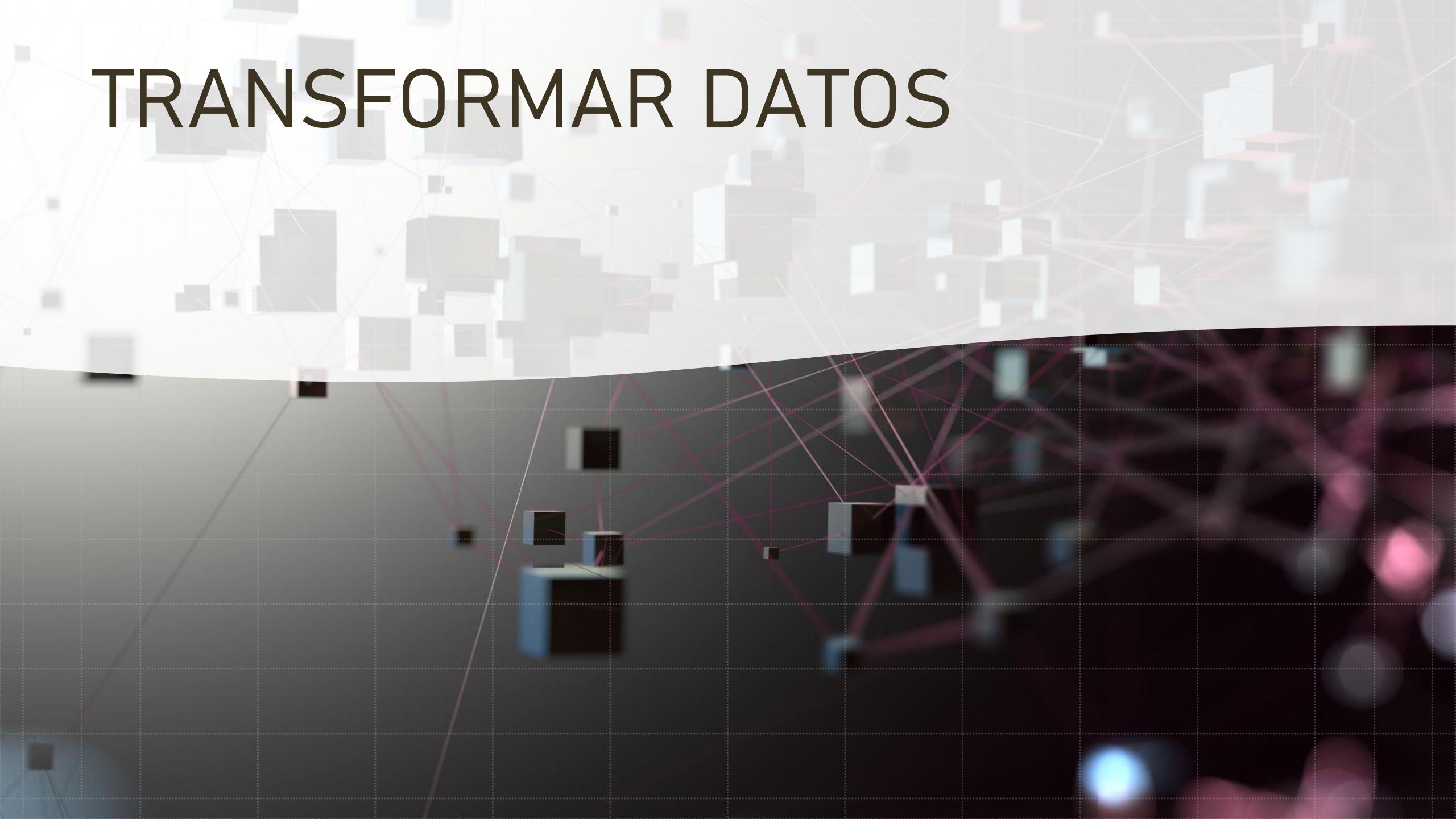
# EJERCICIO

Utilizando la base de datos AgeDataset-V1.csv, realizar calculo con funciones DAX y generar una visualización para identificar:

- ¿Cual es la edad promedio de la muerte de las personas y de que ocupación es más temprana?
- ¿Cuántas diferentes causas de muerte tenemos en el listado?
- ¿Qué diferencia existe en promedio entre la edad de fallecimiento de un político vs un artista?

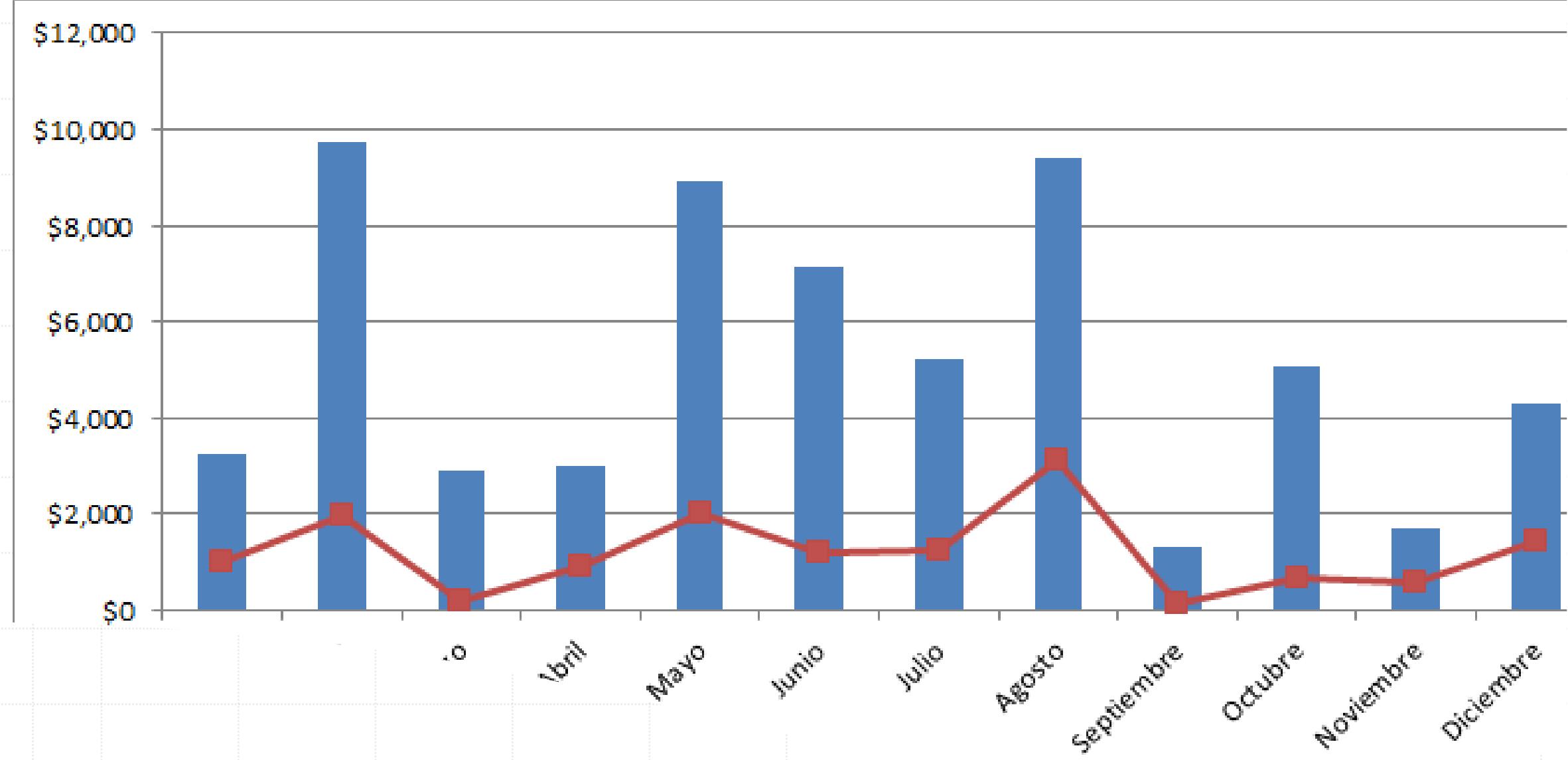


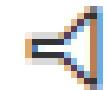
# TRANSFORMAR DATOS



# EJERCICIO

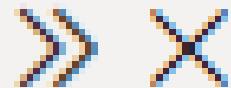
- Utilizando los csv Carpetas Investigación\_FGJCDMX y CAT\_Delitos.
- Eliminar columnas innecesarias (tempo)
- Combinar consultas para obtener la categoría de delito ID\_CatDelito > Tabla CatDelito
- Separar los datos de long/lat en 2 columnas
- Columna condicional para identificar aquellos delitos que son robos





Filters

## Sync slicers

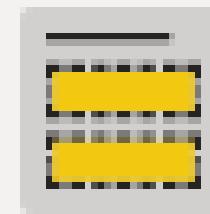


Select a slicer in one of your report pages to start syncing it across other pages

## Visualizations



### Build visual



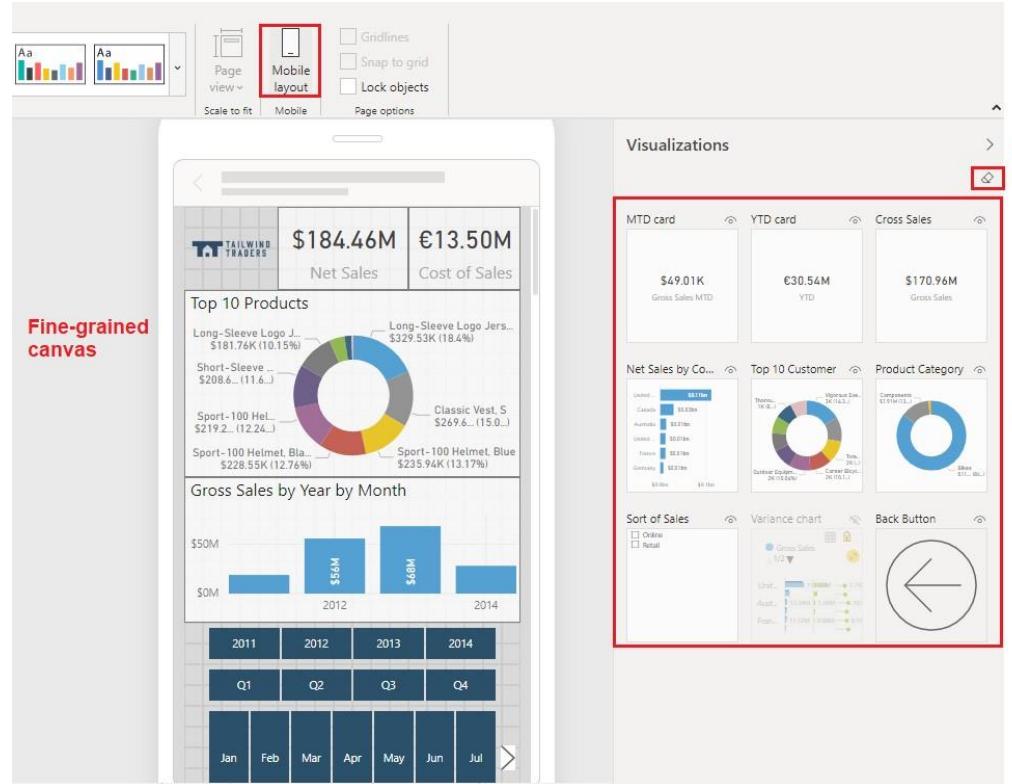
# EJERCICIO



# PUBLICACIÓN



# MÓVILES



# PANELES



# ANCLAR ICONO

Orders over Time

Count of Sales...

0 200

0 20

Anclar al panel

Seleccione un panel existente o cree uno nuevo.

¿Dónde le gustaría anclarlo?

Panel existente

Nuevo panel

Seleccionar el panel existente

Categoría

Anclar Cancelar

# ALERTAS

