Step-by-Step Development of a Business Intelligence Project in Power BI:

The VitalPulse Case

Project carried out by Mario Navarro (independently, using a dataset from Kaggle).

TABLE OF CONTENTS

1. REQUIREMENTS GATHERING

The project began with an in-depth discussion with the client (VitalPulse Sports) to define the business objectives:

- **Improve decision-making** through sales, logistics, product, and customer insights.
- **Optimize inventory and logistics** based on demand trends.
- Focus analysis on **key products and markets.**

Key Metrics: Monthly sales (value and quantity), MoM and YoY comparisons, Trends in orders (delayed, cancelled, average units per order), Customer distribution by state, Top-selling products and categories, Dimensions**,** Time (month, year), Product (name, category, section), Customer (location).

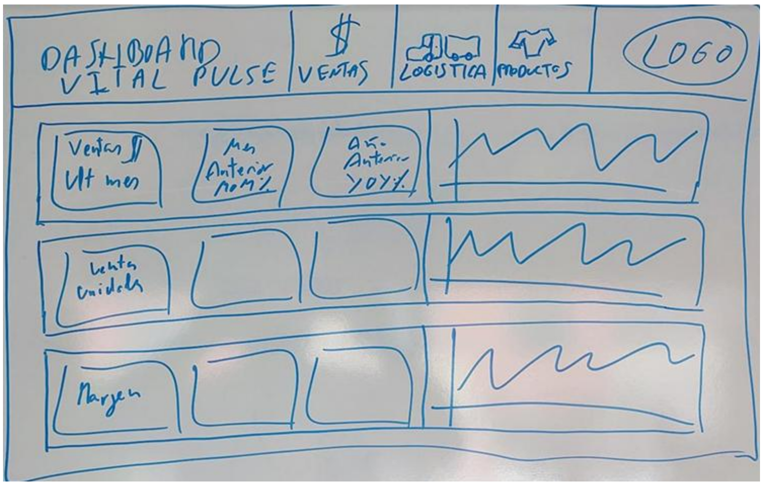
Data Source: ERP system exports (Excel format), with fields covering orders, shipments, products, pricing, margins, and customer details.

Calculations Needed: Sales growth rates, margin evolution, delayed/cancelled orders, six-month and twelve-month trends, and customer segmentation.

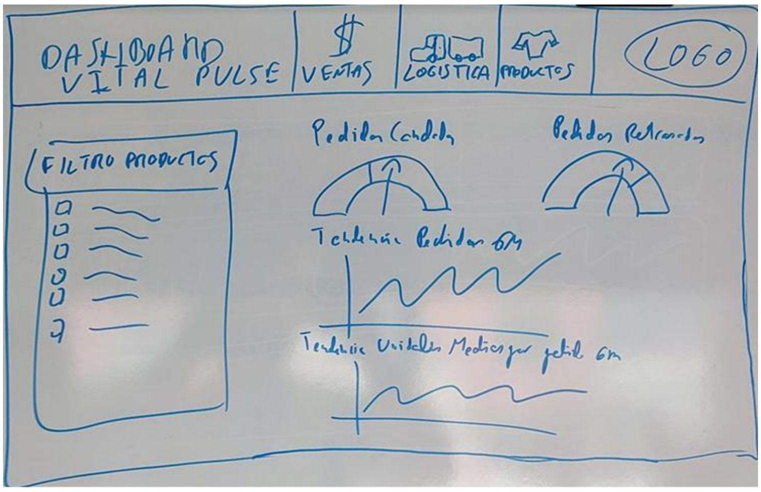
Dashboard Components: KPIs for quick insights, Line charts for trends, Gauges for targets, Maps for customer distribution, Slicers and filters for product, section, geography, and time

We also created a **mockup** to define the structure and agreed on building three reports:

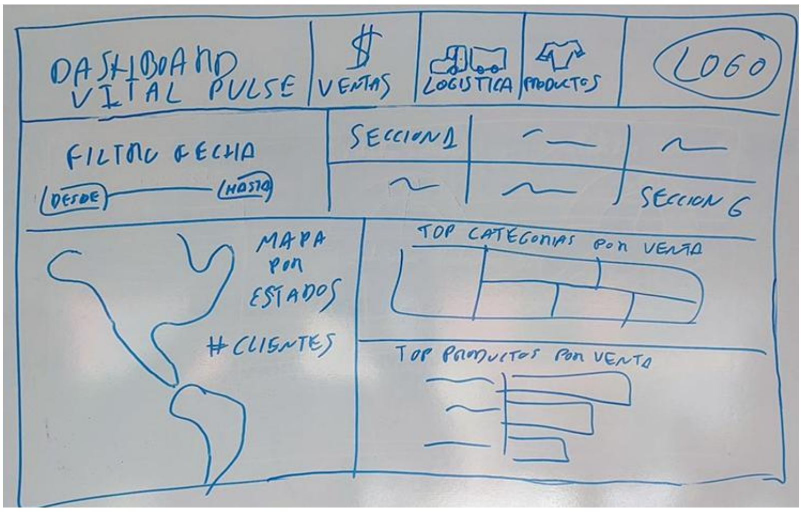
- **Sales**



- **Logistics**



- **Products & Customers**



2. EXTRACTION AND TRANSFORMATION OF DATA

Extraction

- The dataset was exported from the ERP system as a single Excel file with 33 columns in a **flat, denormalized format**
- The table was structured at the level of Sale ID, with each row representing a product within that sale. Since one sale may include multiple orders, the Order ID column appeared repeatedly across rows
- This structure was loaded into Power BI Desktop as the foundation for further preparation

Data Transformation

- Using Power Query, I began shaping the dataset to improve quality and usability: Corrected **data types** (dates, numeric values, and text fields).
- Reviewed the dataset with the Column **Quality, Distribution, and Profile views** to identify duplicates, inconsistencies, and anomalies.
- Detected and **handled null or empty fields**, ensuring the data was clean and ready for modelling

```
= Table.TransformColumnTypes(#"Columnas quitadas",{{" Ciudad del cliente", type text}, {" Estado del cliente", typ
```

	ID venta	ID pedido	Fecha del pedido	Días de envío programado	Fecha de envío
1	1	1	01/01/2021	4	03/01/2021
2	2	2	01/01/2021	4	04/01/2021
3	3	2	01/01/2021	4	04/01/2021
4	4	2	01/01/2021	4	04/01/2021
5	5	4	01/01/2021	4	06/01/2021
6	6	4	01/01/2021	4	06/01/2021
7	7	4	01/01/2021	4	06/01/2021
8	8	4	01/01/2021	4	06/01/2021
9	9	5	01/01/2021	4	07/01/2021
10	10	5	01/01/2021	4	07/01/2021
11	11	5	01/01/2021	4	07/01/2021
12	12	5	01/01/2021	4	07/01/2021
13	13	5	01/01/2021	4	07/01/2021

3. **DATA MODELLING**

Once the dataset was cleaned, I classified all fields into two groups:

- **Metrics** (numerical values for calculations) → sales amount, quantity, discounts, margins, etc.
- **Attributes** (descriptive information) → product details, order status, customer location, categories, etc.

After this classification, I mapped each field to either a **fact table** or a **dimension table**.

With that classification, I mapped fields to fact or dimension tables, guided by granularity:

- An **order groups multiple sales lines** (products).
- Therefore, **shipping-related fields** (e.g., shipping date, planned vs. actual shipping days) live at the **order level**.
- **Price-related metrics** (unit price, totals, discounts, margins) live at the **product line (sale) level** to avoid duplication and keep calculations efficient.

Variable	M or A	FACTS	DIMENSIONS
ID venta	M	f_nvl_ventas	
ID pedido	M	f_nvl_pedido/f_nvl_ventas	
Fecha del pedido	M	f_nvl_pedido/f_nvl_ventas	
Días de envío programado	M	f_nvl_pedido	
Fecha de envío	M	f_nvl_pedido/f_nvl_ventas	
Días de envío real	M	f_nvl_pedido	
Retrasado	A		d_pedido
Estado del pedido	A		d_pedido
Estado de entrega	A		d_pedido
Tipo transaccion	A		d_pedido
Modo de envío	A		d_pedido
ID producto	M	f_nvl_ventas	
Nombre del producto	A		d_productos
Precio del producto	M	f_nvl_ventas	
Cantidad	M	f_nvl_ventas	
Precio total sin descuento	M	f_nvl_ventas	
Tasa de descuento	M	f_nvl_ventas	
Descuento	M	f_nvl_ventas	
Precio total con descuento	M	f_nvl_ventas	
Margen %	M	f_nvl_ventas	
Margen $	M	f_nvl_ventas	
ID categoría	A		d_productos
Nombre de categoría	A		d_productos
ID seccion	A		d_productos
Nombre de seccion	A		d_productos
ID cliente	M	f_nvl_pedido/f_nvl_ventas	
Segmento del cliente	A		d_clientes
Ciudad del cliente	A		d_clientes
Estado del cliente	A		d_clientes
País del cliente	A		d_clientes

I then designed a star schema and documented it with fact tables at the bottom and dimensions at the top:

<u>Fact tables</u>

- **f_nvl_ventas** (sales line–level): Sale ID, Order ID, Product ID, dates at line scope, quantity, unit price, totals, discounts, margin %, margin $.
- **f_nvl_pedido** (order-level): Order ID, Customer ID, order date, shipping date, planned vs. actual shipping days.

<u>Dimension tables</u>

- **d_clientes**: segment, city, state, country.
- **d_pedido**: order status, delivery status, transaction type, shipping mode.
- **d_productos**: product name, category, section.
- **d_fechas**: date hierarchy for time intelligence.

This setup enforces the right grain (orders vs. sales lines), keeps price metrics at the line where they're generated, and places shipping events where they occur (order). The result is a performant model with clean relationships and reliable calculations.

d_clientes
ID cliente
Segmento del cliente
Ciudad del cliente
Estado del cliente
País del cliente

d_pedido
ID pedido
Retrasado
Estado del pedido
Estado de entrega
Tipo transaccion
Modo de envío

d_productos
ID producto
ID categoría
ID seccion
Nombre del producto
Nombre de categoría
Nombre de seccion

d_fechas

f_nvl_ventas
ID venta
ID cliente
ID pedido
ID producto
Fecha del pedido
Fecha de envío
Precio del producto
Cantidad
Precio total sin descuento
Tasa de descuento
Descuento
Precio total con descuento
Margen %
Margen $

f_nvl_pedido
ID pedido
ID cliente
Fecha del pedido
Fecha de envío
Días de envío programado
Días de envío real

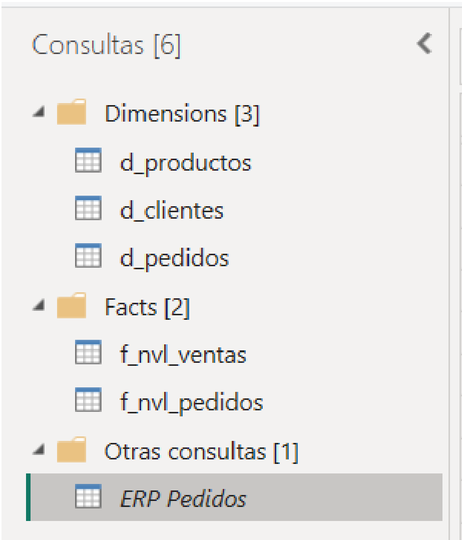
With the data model defined, the next step was to create the **fact and dimension tables** in Power Query. To do this, I referenced the original ERP table as many times as needed to generate each new table.

For every fact or dimension table, I applied a **structured process:**

- Keep only the relevant fields for that table.
- Sort by the primary key (or create one if it didn't exist).
- Remove null values in key fields.
- Remove duplicates to ensure integrity.

Since all dimension tables already had **natural keys** (e.g., Product ID, Customer ID, Order ID), it was **not necessary to create surrogate keys** or combined columns in the fact tables for linking.

Additionally, I **disabled load** for the original source table, as it was only used as a reference to generate the others.



Example: clients table

	ID cliente	Segmento del cliente	Ciudad del cliente	Estado del cliente	País del cliente
1	1	Consumer	Brownsville	TX	EE. UU.
2	2	Consumer	Littleton	CO	EE. UU.
3	3	Consumer	Caguas	PR	Puerto Rico
4	4	Consumer	San Marcos	CA	EE. UU.
5	5	Home Office	Caguas	PR	Puerto Rico
6	6	Consumer	Passaic	NJ	EE. UU.
7	7	Corporate	Caguas	PR	Puerto Rico
8	8	Corporate	Lawrence	MA	EE. UU.
9	9	Consumer	Caguas	PR	Puerto Rico
10	10	Corporate	Stafford	VA	EE. UU.
11	11	Consumer	Caguas	PR	Puerto Rico
12	12	Corporate	San Antonio	TX	EE. UU.
13	13	Home Office	Caguas	PR	Puerto Rico
14	14	Corporate	Pico Rivera	CA	EE. UU.
15	15	Corporate	Fontana	CA	EE. UU.
16	16	Corporate	Caguas	PR	Puerto Rico
17	17	Consumer	Taylor	MI	EE. UU.
18	18	Consumer	Martinez	CA	EE. UU.
19	19	Home Office	Caguas	PR	Puerto Rico
20	20	Consumer	West New York	NJ	EE. UU.
21	21	Consumer	Caguas	PR	Puerto Rico

After creating all the tables, I created the missing date dimension directly in Power BI using DAX with the CALENDAR function, enabling full time intelligence across the model.

```
d_fechas =
VAR BaseCalendar=
 CALENDARAUTO()
 RETURN
    SELECTCOLUMNS (
        BaseCalendar,
        "Fecha", [Date],
        "Año", YEAR ( [Date] ),
        "Trimestre", QUARTER( [Date] ),
        "MesNum", MONTH ( [Date] ),
        "MesNombre", FORMAT ( [Date], "mmmm" ),
        "AnoMesNombre", FORMAT ( [Date], "mmm yy" ),
        "AnoMesNum", YEAR ( [Date] ) * 100 + MONTH ( [Date] ),
        "NumSemanaAno", WEEKNUM( [Date], 1 ),
        "DiaMes", DAY( [Date] ),
        "DiaSemanaNum", WEEKDAY( [Date] ),
        "DiaSemanaNombre", FORMAT ( [Date], "dddd" )
    )
```

Fecha	Año	Trimestre	MesNum	MesNombre	AnoMesNombre	AnoMesNum	NumSemanaAno	DiaMes	DiaSemanaNum	DiaSemanaNombre
01/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	1	1	6	viernes
02/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	1	2	7	sábado
03/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	2	3	1	domingo
04/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	2	4	2	lunes
05/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	2	5	3	martes
06/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	2	6	4	miércoles
07/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	2	7	5	jueves
08/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	2	8	6	viernes
09/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	2	9	7	sábado
10/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	3	10	1	domingo
11/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	3	11	2	lunes
12/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	3	12	3	martes
13/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	3	13	4	miércoles
14/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	3	14	5	jueves
15/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	3	15	6	viernes
16/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	3	16	7	sábado
17/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	4	17	1	domingo
18/01/2021 0:00:00	2021	1	1	enero	ene 21	202101	4	18	2	lunes

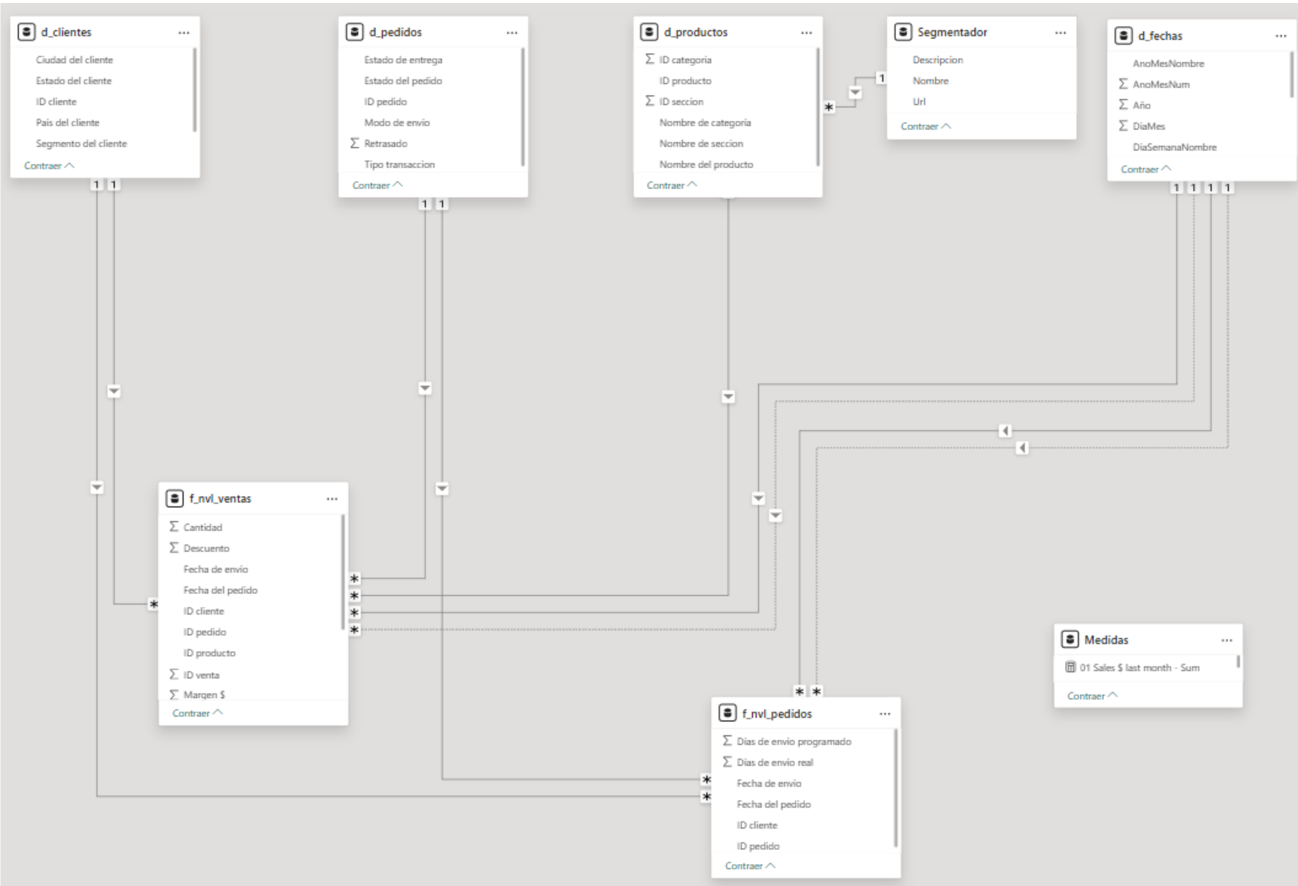
After creating all the fact and dimension tables in Power Query, I moved to the Model view in Power BI to **establish relationships**.

The final model follows the **same structure designed in the initial mock model**, ensuring consistency between the planning phase and the implementation.

<u>Relationships:</u>

- **One-to-many links** were established from each dimension to the corresponding fact tables using their primary keys.
- For the **date dimension**, the relationship with Order Date was set as active, while Shipping Date remained inactive, allowing flexible analysis of sales vs. logistics timelines.

This schema ensures efficiency by keeping metrics at the right grain (order vs. product line) and makes the model intuitive for building dashboards.



4. **COMPUTATIONS AND DAX MEASURES**

After analyzing the business requirements and discussing the objectives with the client during our first meeting, we concluded that the dashboards would require a well-defined set of metrics to provide actionable insights. These metrics were carefully selected to cover sales performance, logistics efficiency, product analysis, and customer distribution. The **finalized list of metrics** includes:

<u>Measures for the sales dashboard</u>

01_Sales $ last month – Sum

02_ Sales $ previous month – Sum

03_ Sales $ previous year - Sum

04_ % MoM $ Sales

05_ Arrow % MoM $

06_ % YoY $ Sales

07_ Arrow % YoY $

08_Trend Ventas $ Last 12 months

09_Margin % - Avg

10_ Margin % Last Month - Avg

11_ Margin % Previous Month - Avg

12_ Margin % Same Month Previous Year

13_MoM% Margin %

14_Arrow MoM% Margin %

15_YoY % Margen %

16_Arrow YoY% Margin %

17_Trend Margin % Last 12 Meses

18_Sales Units Last Mes - Sum

19_ Sales Units Previous Month - Sum

20_ Sales Units Same Month Previous Year - Sum

21_MoM % Sales Units

22_Arrow MoM% Sales Units

23_YoY % Sales Units

24_Arrow YoY% Sales Units

25_Trend Sales Units Last 12 Meses

<u>Measures for the orders and logistics dashboard</u>

26_Number of cancelled orders

27_YtD Cancelled orders

28_Number of delayed Orders

29_YtD Delayed Orders

30_Orders last 6 Months

31_Average Units Sold per Order

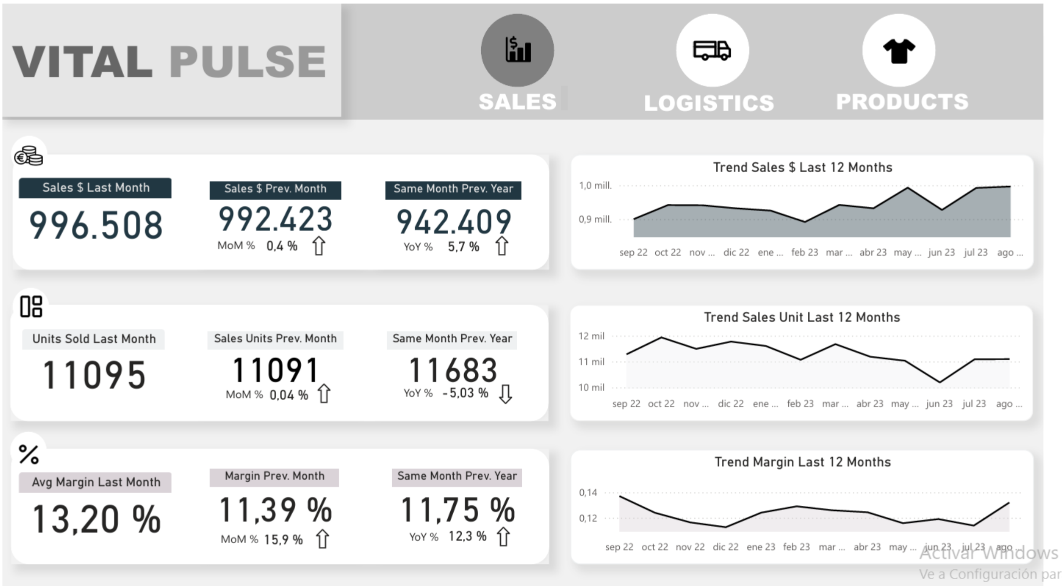
32_Units sold per Order last 6 Months

<u>Measures for the clients and products dashboard</u>

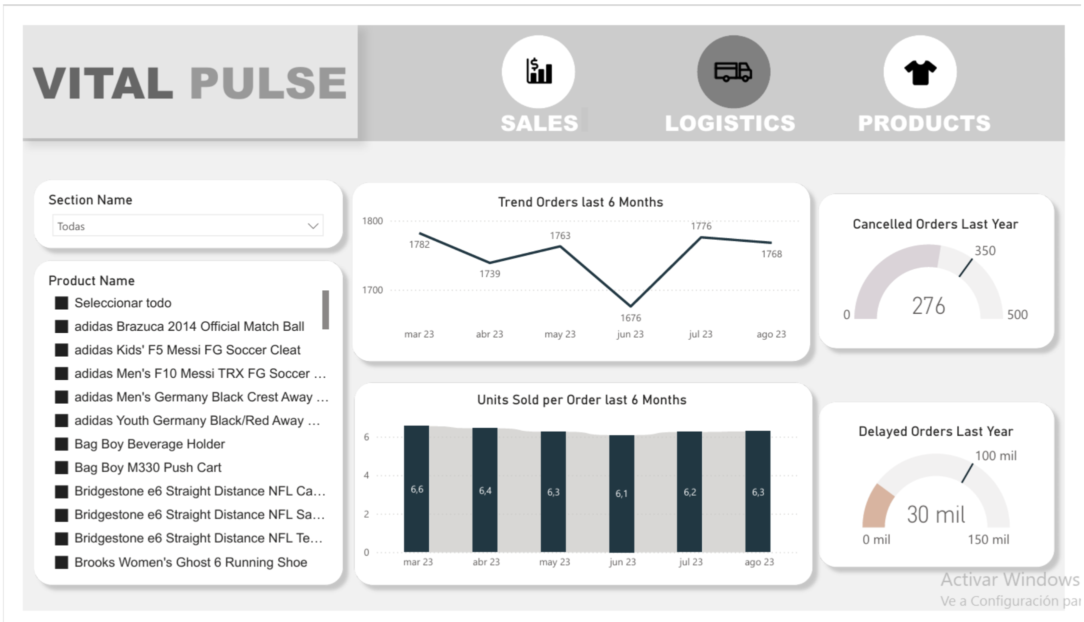
33_Number of clients

5. **DASHBOARD DEVELOPMENT**

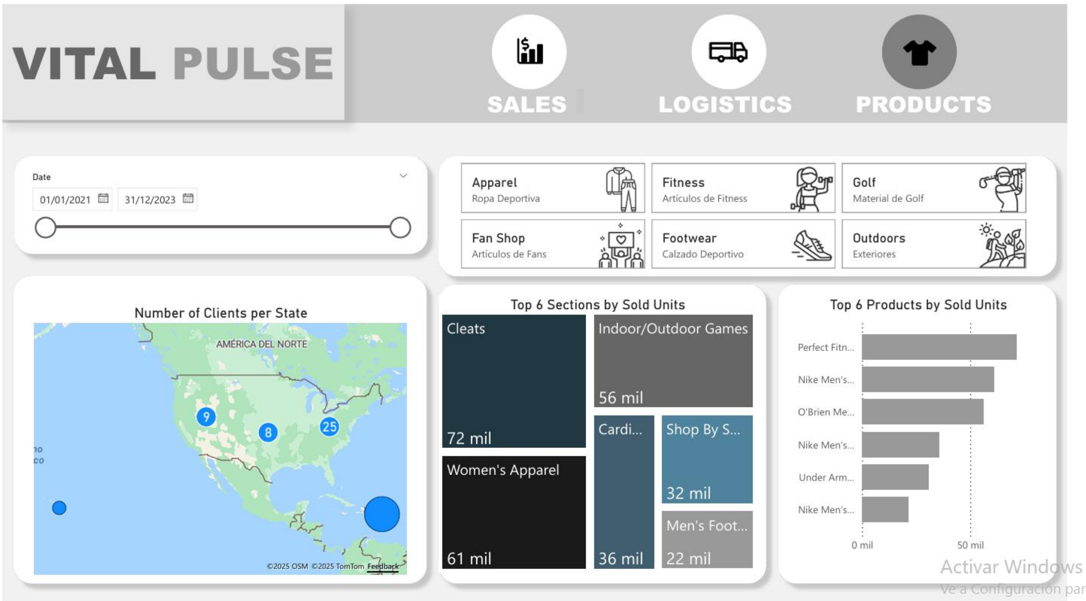
The dashboard was structured into **three main sections**: Sales, Logistics, and Products & Customers, to align with the client's key objectives. In the **Sales dashboard,** KPI cards display monthly revenue, units sold, and margins, with MoM and YoY comparisons supported by arrows for quick interpretation. Line charts illustrate 12-month trends, providing a clear view of sales performance over time.



The **Logistics dashboard** focuses on operational efficiency, including cancelled and delayed orders, order trends for the last six months, and the average units per order. These visuals highlight potential bottlenecks in delivery and fulfilment.



The **Products & Customers dashboard** includes a geographic map of clients by state, along with rankings of top-selling products and categories. Filters and slicers allow users to drill down by product, section, and date. A specific challenge was correcting the misclassification of Puerto Rico in the map, which was solved by combining state and country fields.



Overall, the dashboards were designed with interactivity, clarity, and usability in mind, ensuring stakeholders can explore the data and derive insights efficiently.

6. PUBLISHING AND SHARING

Once finalized, the report was published to the Power BI Service, ensuring secure access for stakeholders. A public link and an embedded iframe were generated to allow easy sharing via email and integration into external websites. This step guaranteed that the dashboards were not only functional but also widely accessible for decision-makers.

- Link that can be shared via email: https://app.powerbi.com/view?r=eyJrIjoiOGI4MzllNjQtMWY0Yi00MmM1LWI0NDMtMDUwMmVmODIzMzVhIiwidCI6IjAzYTBmYjY5LWE0ZDAtNDQyZC1hNGQ0LWNmYjVkYTgwNzUwMCJ9

- HTML to embed on a website:

<iframe title="Proyecto final" width="600" height="373.5" src="https://app.powerbi.com/view?r=eyJrIjoiOGI4MzllNjQtMWY0Yi00MmM1LWI0NDMtMDUwMmVmODIzMzVhIiwidCI6IjAzYTBmYjY5LWE0ZDAtNDQyZC1hNGQ0LWNmYjVkYTgwNzUwMCJ9" frameborder="0" allowFullScreen="true"></iframe>