

21 DE ENERO DE 2026



TAREA 4.4 - ORGANIZACIÓN DE INFORMES CON DATAPANE

REALIZADO POR DANIEL GARCÍA MÉNDEZ

Contenido

Explicación de la organización de los componentes del informe:.....	2
Contenido incluido en los selectores:.....	8
Webgrafía.....	10

Explicación de la organización de los componentes del informe:

¿Qué he incluido en los grupos?

En la página Dashboard incluí `dp.Group` para poner indicadores usando `BigNumber` en dos columnas:

- Una para las ventas totales de los años 2017-2021
- Ventas totales en 2021

Lo puse así porque son elementos que se relacionan, por lo que resultan legibles fácilmente. Además, permite comparar las ventas totales con las de 2021.

Ejemplo de código:

```
# 2. CÁLCULO DE INDICADORES

# Ventas totales en todo el periodo
total_ventas = df["ventas"].sum()

# Ventas totales en el último año (2021)
ventas_2021 = df[df["año"] == 2021]["ventas"].sum()
```

Captura de visualización de informe de estos dos elementos:

Resumen de Ventas

Ventas Totales (2017-2021)

122,410 €

Ventas en 2021

23,227 €

También puse un indicador usando `BigNumber` para mostrar la categoría del producto más vendido y su total de ventas.

La primera línea de código agrupa los datos por tipo de producto y calcula la suma total de ventas para cada uno usando el método `.sum()`. A continuación, se identifica el producto con mayor volumen de ventas mediante `.idxmax()`, que devuelve la clave (nombre del producto) asociado al valor máximo. Por último, se obtiene ese valor máximo con `.max()`, que representa el total de ventas del producto más vendido.

```
# Producto con más ventas totales
ventas_por_producto = df.groupby("tipo de producto")["ventas"].sum()
producto_top = ventas_por_producto.idxmax()
ventas_producto_top = ventas_por_producto.max()
```

Captura de visualización de comparativa años + producto más vendido:

Dashboard Comparativas Datos

Resumen de Ventas

Ventas Totales (2017-2021)

122,410 €

Ventas en 2021

23,227 €

Producto más vendido (total periodo)

Moda (43,118 €)

Componentes separados en páginas

Página 1 Dashboard

En esta incluí:

- Ventas totales 2017-2021
- Ventas 2021
- Producto más vendido junto a su categoría y total de ventas
- Gráfico con la evolución total de ventas

Captura de visualización página 1 dashboard:



Captura del código con gráfico:

Aquí incluí un texto con el encabezado, indicadores clave de ventas, producto más vendido y el gráfico.

```
# Página 1: Dashboard Principal con indicadores y gráfico
pagina1 = dp.Page([
    title="Dashboard",
    blocks=[
        dp.Text("# Resumen de Ventas"),
        dp.Group(
            dp.BigNumber(heading="Ventas Totales (2017-2021)", value=f"{total_ventas:,.0f} €"),
            dp.BigNumber(heading="Ventas en 2021", value=f"{ventas_2021:,.0f} €"),
            columns=2
        ),
        dp.BigNumber(
            heading="Producto más vendido (total periodo)",
            value=f"{producto_top} ({ventas_producto_top:,.0f} €)"
        ),
        bloque_linea
    ]
])
```

Lo puse así ya que las comparativas generan mucha información visual. Separarlas en una página propia evita saturar el dashboard y permite al usuario explorar cada vista sin distracciones.

Página 2 Comparativas

En esta página incluí:

Un selector con dos gráficos comparativos:

- Barras por año y Producto
- Barras por Región y Año

Lo puse así ya que Las comparativas generan mucha información visual. Separarlas en una página propia evita saturar el dashboard y permite al usuario explorar cada vista sin distracciones.

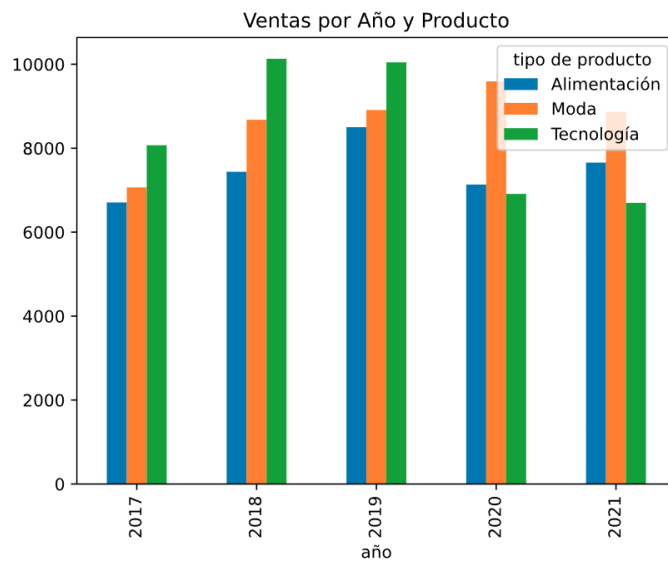
Captura de visualización de página 2 con los gráficos:

```
# --- Gráfico 1: Barras apiladas por AÑO y TIPO DE PRODUCTO ---  
plt.figure()  
ventas_año_producto = df.groupby(["año", "tipo de producto"])["ventas"].sum().unstack()  
plot1 = ventas_año_producto.plot(kind="bar", title="Ventas por Año y Producto")  
bloque_bar_apilado = dp.Plot(plot1.get_figure(), label="Año vs Producto")
```

Dashboard Comparativas Datos

Comparativas por Año, Producto y Región

Año vs Producto Región vs Año

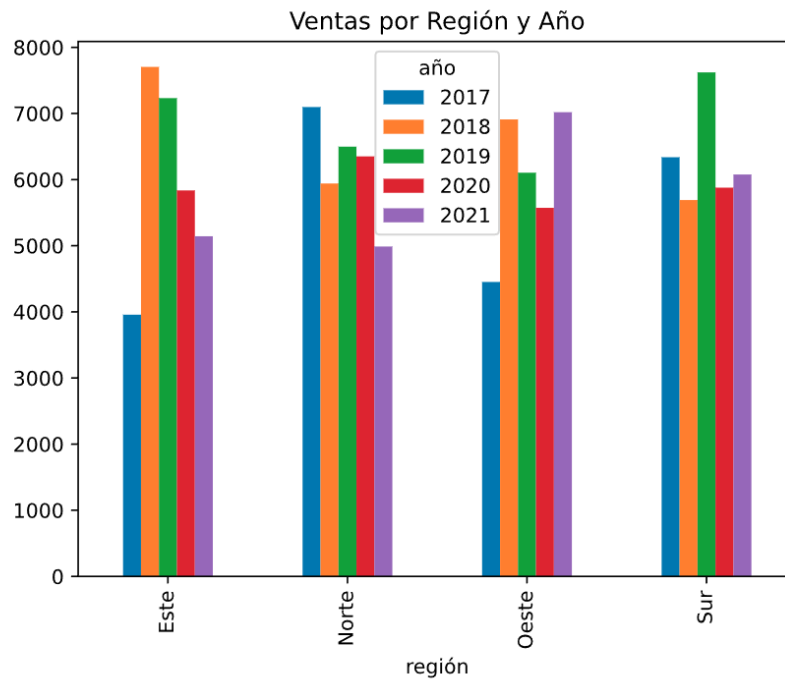


```
# --- Gráfico 3: Líneas de evolución total por AÑO ---  
plt.figure()  
ventas_año = df.groupby("año")["ventas"].sum()  
plot3 = ventas_año.plot(kind="line", marker="o", title="Evolución Total de Ventas")  
bloque_linea = dp.Plot(plot3.get_figure(), label="Evolución Temporal")
```

Dashboard Comparativas Datos

Comparativas por Año, Producto y Región

Año vs Producto Región vs Año



Página 3 Datos

En esta página incluí una tabla interactiva de todos los datos del csv, lo incluí ya que es útil por si el usuario quiere hacer algún tipo de filtrado u ordenación personalizados.

Captura de visualización de la tabla interactiva

Dashboard
Comparativas
Datos

Tabla Completa de Registros

60 rows
4 columns
240 cells
Run SQL Query
Export

	año	región	tipo de producto	ventas
1	2017	Norte	Tecnología	1980
2	2017	Sur	Tecnología	2642
3	2017	Este	Tecnología	1594
4	2017	Oeste	Tecnología	1855
5	2017	Norte	Moda	2687
6	2017	Sur	Moda	1824
7	2017	Este	Moda	1297
8	2017	Oeste	Moda	1258
9	2017	Norte	Alimentación	2430
10	2017	Sur	Alimentación	1872
11	2017	Este	Alimentación	1066

Contenido incluido en los selectores:

Aquí incluí un título y un bloque con texto en negrita y otros bloques de selección entre los dos gráficos

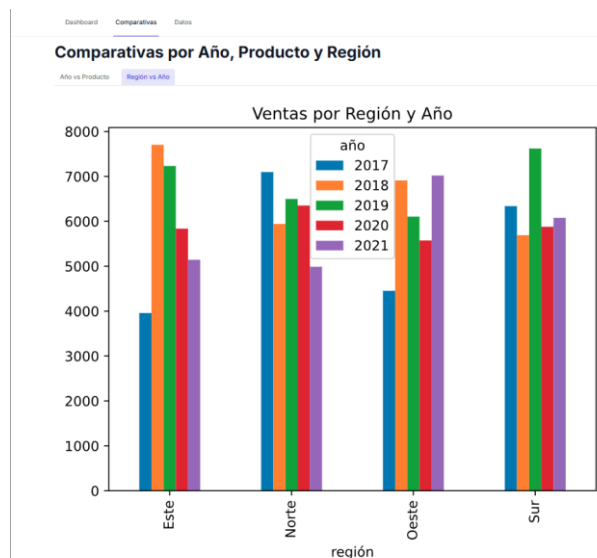
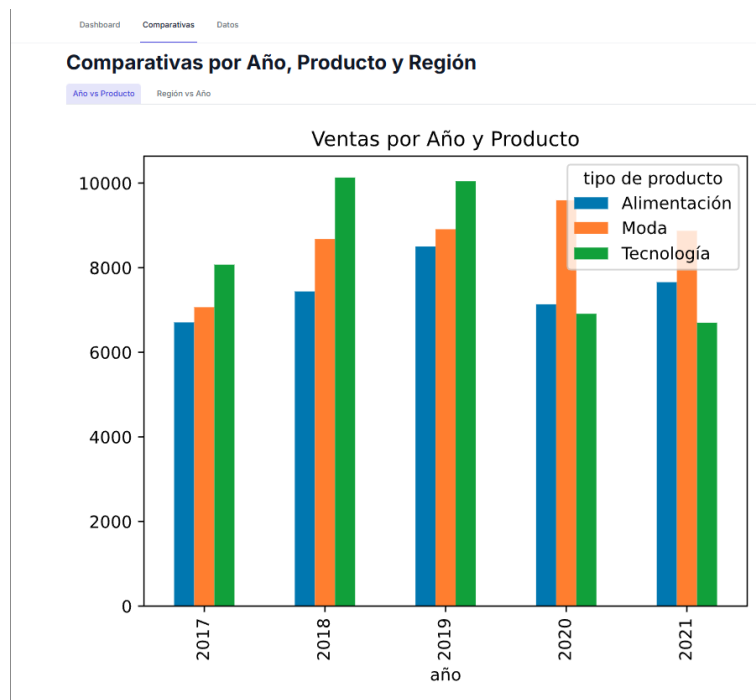
```
# Página 2: Comparativas con Selectores
pagina2 = dp.Page(
    title="Comparativas",
    blocks=[
        dp.Text("# Comparativas por Año, Producto y Región"),
        dp.Select(
            blocks=[
                bloque_bar_apilado,
                bloque_bar_region
            ]
        )
    ]
)
```

Código de gráficos

```
# --- Gráfico 1: Barras apiladas por AÑO y TIPO DE PRODUCTO ---
plt.figure()
ventas_año_producto = df.groupby(["año", "tipo de producto"])[ "ventas"].sum().unstack()
plot1 = ventas_año_producto.plot(kind="bar", title="Ventas por Año y Producto")
bloque_bar_apilado = dp.Plot(plot1.get_figure(), label="Año vs Producto")

# --- Gráfico 2: Barras por REGIÓN y AÑO ---
plt.figure()
ventas_region_año = df.groupby(["región", "año"])[ "ventas"].sum().unstack()
plot2 = ventas_region_año.plot(kind="bar", title="Ventas por Región y Año")
bloque_bar_region = dp.Plot(plot2.get_figure(), label="Región vs Año")
```

Captura de visualización de las dos pestañas:



Webgrafía

<https://docs.datapane.com>

<https://github.com/datapane/datapane>