

14 DE ENERO DE 2026



## TAREA 4.2 - INFORMES INTERACTIVOS CON DATAPANE

**Nombre:** Daniel García Méndez

**Curso:** 2ºDAM

**Módulo:** Desarrollo de interfaces

**Unidad:** Tema 4

## Contenido

TAREA 4.2 - Informes interactivos con Datapane .....	3
Ejercicio 1. Filtrado y análisis de ventas .....	3
Ejercicio 2. Diseño de un informe de ventas .....	8
Webgrafía.....	12

## TAREA 4.2 - Informes interactivos con Datapane

### Contexto

En esta tarea vas a trabajar con informes interactivos dentro de una aplicación, utilizando Python, pandas y la librería Datapane.

El objetivo es que aprendas a:

- **Analizar datos** a partir de ficheros CSV.
- **Diseñar informes** que combinen tablas interactivas e indicadores visuales.
- **Utilizar correctamente** los componentes DataTable, Table y BigNumber.

### Ejercicio 1. Filtrado y análisis de ventas

Partiendo del informe DI\_U05\_A02\_03.html visto en clase y del fichero CSV proporcionado en los apuntes, debes profundizar en las funcionalidades de filtrado y ordenación del componente DataTable.

Obten, a partir de los datos, la siguiente información:

1. Los vendedores que en el mes de julio han vendido más de 500 unidades.

Este comando sirve para quedarme solo con los vendedores de la columna mes perteneciente a Julio y los datos de la columna unidades mayor que 500.

```
# 1. Vendedores en julio con >500 unidades
ej1_julio = df[(df['Mes'] == 'Julio') & (df['Unidades'] > 500)]
```

Luego usamos BigNumber para mostrar el resultado en grande dentro del informe.

La línea len(ej1\_julio) cuenta cuántos vendedores hay en ese filtro.

Le añado "Unidades" para que se entienda mejor visualmente.

El número que aparece es el total de vendedores que cumplen lo anterior.

```
big_julio = dp.BigNumber(
    heading="Vendedores en julio con >500 unidades",
    value=f"{len(ej1_julio)} Unidades"
)
```

**2. El vendedor con mayor importe de ventas en un mes, sin superar las 300 unidades, indicando también el mesen el que lo consiguió.**

Este comando sirve para quedarme solo con las unidades menor o igual a 300, usé la función `sort_values` incluida en Python para ordenar por la columna `Importe` de manera descendente.

```
# 2. Mayor importe con ≤300 unidades
ej1_importe = df[df['Unidades'] <= 300].sort_values(by="Importe", ascending=False).head(1)
```

Después uso `BigNumber` para mostrar el resultado del filtrado con un texto de cabecera que queramos, en mi caso es "Importe máximo con ≤300 unidades"

```
big_importe = dp.BigNumber(
    heading="Importe máximo con ≤300 unidades",
    value=float(ej1_importe["Importe"].iloc[0])
)
```

**3. De los vendedores cuyo nombre empieza por la letra S, identificar quién vendió más unidades en el mes de agosto.**

Uso esta instrucción para seleccionar la columna `mes` el mes de agosto y de la columna `nombre` solo los nombres que empiezan por S, para eso usé la función del tipo de dato `str` `startswith("S")`.

```
# 3. Vendedores que empiezan por S en agosto
ej1_agosto_s = df[(df["Mes"] == "Agosto") & (df["Nombre"].str.startswith("S"))]
ej1_agosto_s = ej1_agosto_s.sort_values(by="Unidades", ascending=False).head(1)
```

Después uso el `BigNumber` para mostrar en grande un texto a elegir y los valores

```
✓ big_agosto_s = dp.BigNumber(
    heading="Máximo vendedor 'S' en agosto",
    value=int(ej1_agosto_s["Unidades"].iloc[0])
)
```

También uní todos los filtros usados con Report para ver visualmente la tabla con los datos específicos.

```
texto = dp.Text("***Resultados obtenidos aplicando filtros y ordenaciones con DataTable.**")
fichero = dp.Attachment(file='DI_U05_A02_02.csv')

# 4) INFORME ÚNICO FINAL (SOLO FILTROS) AQUÍ SALEN LAS TABLAS CON TODOS LOS FILTROS APLICADOS.

report = dp.Report(
    titulo,
    dp.Group(big_julio, big_importe, big_agosto_s, columns=3),

    dp.Text("### Resultado 1: Julio > 500 unidades"),
    tabla_julio,

    dp.Text("### Resultado 2: Importe máximo con ≤300 unidades"),
    tabla_importe,

    dp.Text("### Resultado 3: Vendedores que empiezan por S en agosto"),
    tabla_agosto_s,

    texto,
    fichero
)

report.save(path="DI_U05_A02_03_EJ1_FINAL.html", open=True)
```

**Captura de pantalla del resultado de cada filtro:**

Ejercicio 1 – Filtrado y análisis de ventas		
Vendedores en julio con >500 unidades <b>2 Unidades</b>	Importe máximo con ≤300 unidades <b>5062.0</b>	Máximo vendedor 'S' en agosto <b>504</b>

**Captura de pantalla con una query completa:**

## TAREA 4.2 - INFORMES INTERACTIVOS CON DATAPANE

### Resultado 1: Julio > 500 unidades

2 rows		4 columns	8 cells	Run SQL Query	Export ▾
	Nombre	Mes	Unidades	Importe	
1	Stephanie Cummings	Julio	533	4761	
2	Priscilla Hull	Julio	550	4998	

### Resultado 2: Importe máximo con ≤300 unidades

1 rows		4 columns	4 cells	Run SQL Query	Export ▾
	Nombre	Mes	Unidades	Importe	
1	Angelica Armstrong	Noviembre	188	5062	

### Resultado 3: Vendedores que empiezan por S en agosto

1 rows		4 columns	4 cells	Run SQL Query	Export ▾
	Nombre	Mes	Unidades	Importe	
1	Stephanie Cummings	Agosto	504	2548	

### Indicaciones

- Utiliza las opciones de filtrado y ordenación de la tabla interactiva.
- Para cada apartado:
  - Indica qué filtros y ordenaciones has utilizado.
  - Incluye una captura donde se vea claramente el resultado final.

Una vez obtenidos los resultados:

- **Explica brevemente por qué el uso de una tabla interactiva es adecuado para este tipo de consultas.**

El perfil que más partido le saca es alguien que **necesita ver datos rápido y tomar decisiones**:

- **Dirección** → para ver cifras claras sin perder tiempo.
- **Responsable comercial** → para revisar ventas, clientes o estados al vuelo.
- **Analista** → porque puede cruzar datos sin tener que exportarlos.

- **Indica qué perfil de usuario (dirección, responsable comercial, analista...) se beneficiaría más de este tipo de informe.**

El perfil que más partido le saca es alguien que **necesita ver datos rápido y tomar decisiones**:

- **Dirección** → para ver cifras claras sin perder tiempo.
- **Responsable comercial** → para revisar ventas, clientes o estados al vuelo.
- **Analista** → porque puede cruzar datos sin tener que exportarlos.

## Ejercicio 2. Diseño de un informe de ventas

En este segundo ejercicio deberás diseñar un informe de ventas para una cadena de grandes almacenes, a partir del fichero:

DI\_U05\_A02\_PP\_E\_01.csv.

El informe debe mostrar información de ventas anuales (2017–2021) por:

- Región (norte, sur, este y oeste)
- Tipo de producto (alimentación, tecnología y moda)

El informe debe incluir obligatoriamente y en este orden:

1. El logotipo de la empresa.
2. El título del informe: Informe de ventas.
3. Una sección inicial de resumen ejecutivo, que incluya:
  - El total de ventas acumuladas de todos los años.
  - El año con mayor volumen de ventas.
  - Un breve texto (2–3 líneas) justificando por qué estos datos son relevantes para la dirección.
4. El importe total de ventas del año 2021 y su comparación con el total del año anterior.

5. Una tabla interactiva (DataTable) con los datos del informe, que permita filtrar y ordenar.

**Captura de pantalla de resultado:**



6. La posibilidad de descargar el fichero con los datos del informe.

Utilizando la tabla interactiva del informe:

- Indica qué región ha tenido mayores ventas en el año 2021.

**Captura de la tabla interactiva con el año 2021 y el máximo de ventas:**

Con el filtro año 2021 y ventas en orden descendente, me sale que el tipo de producto más vendido es Moda en la región Sur y en número de ventas es 2898.

60 rows 4 columns 240 cells						Run SQL Query	Export
	Año	Región	Tipo	Ventas			
1	2021	Sur	Moda	2898			
2	2021	Oeste	Alimentación	2729			
3	2021	Oeste	Tecnología	2334			
4	2021	Este	Moda	2311			
5	2021	Norte	Alimentación	1974			
6	2021	Oeste	Moda	1956			
7	2021	Sur	Alimentación	1882			
8	2021	Este	Tecnología	1760			
9	2021	Norte	Moda	1707			
10	2021	Norte	Tecnología	1307			
11	2021	Sur	Tecnología	1207			

- Indica qué tipo de producto ha generado más ventas en el conjunto de todos los años.

**Captura de la tabla interactiva con el producto que ha generado más ventas en conjunto de todos los años.**

El tipo de producto que más ventas ha generado es moda en la región Sur.

60 rows 4 columns 240 cells						Run SQL Query	Export
	Año	Región	Tipo	Ventas			
1	2020	Sur	Moda	2962			
2	2018	Este	Moda	2952			
3	2020	Norte	Moda	2937			
4	2021	Sur	Moda	2898			
5	2018	Norte	Tecnología	2809			
6	2019	Sur	Tecnología	2793			
7	2021	Oeste	Alimentación	2729			
8	2019	Oeste	Tecnología	2725			
9	2019	Este	Moda	2723			
10	2017	Norte	Moda	2687			
11	2017	Sur	Tecnología	2642			

Explica qué filtros y ordenaciones has utilizado para llegar a cada conclusión.

**Total de ventas acumuladas entre los años 2017 y 2021:**

Sumé todas las filas de la columna ventas con el siguiente comando

```
# Total acumulado 2017 - 2021
total_acumulado = df["Ventas"].sum()
```

Después realicé un BigNumber para ver el resultado en grande:

En el BigNumber utilizo como cabecera un texto a elegir y en el valor uso el total acumulado.

```
# BigNumber resumen ejecutivo
resumen = dp.BigNumber(
    heading="Total ventas acumuladas (2017 - 2021)",
    value=total_acumulado,
    change=importe_max,
    is_upward_change=True
)
```

### Año con mayor volumen de ventas:

Para hacer eso, debo hacer el siguiente comando:

```
# Ventas por año
ventas_por_año = df.groupby("Año")["Ventas"].sum()
```

Eso no da una tabla agrupada con el total de ventas de cada año.

Aquí usamos la siguiente instrucción para ordenar las ventas por año

```
# Año con mayor volumen
año_max = ventas_por_año.idxmax()
```

### Comparativa entre 2021 y 2020:

Para esto usaremos las siguientes instrucciones para obtener las ventas de determinados años:

```
# Ventas 2021 y 2020
ventas_2021 = ventas_por_año.get(2021, 0)
ventas_2020 = ventas_por_año.get(2020, 0)
```

Usaremos esta instrucción para obtener la diferencia.

```
diferencia = ventas_2021 - ventas_2020
```

### Tabla interactiva:

Para realizar eso, debo leer la tabla del csv para luego realizar un reporte incluyendo esta.

```
data_table = dp.DataTable(df)
```

En la imagen señalada se vería la tabla interactiva.

```
# Reporte final
report = dp.Report(
    imagen,
    titulo,
    dp.HTML("<h2>Resumen ejecutivo</h2>"),
    resumen,
    comparativa,
    anioMaxVolumen,
    texto_resumen,
    data_table,
    fichero
)
```

Para realizar eso, no usé ningún filtrado, tampoco una ordenación porque no es necesario, el usuario podrá establecer sus propios criterios de ordenación.

Para adjuntar el csv para descargar, puse el siguiente código:

```
# Adjuntar CSV para descargar
fichero = dp.Attachment(file=fichero_csv)
```

Webgrafía

<https://experts-deny-b9a.craft.me/5EVzmnvFZAgHAP>