

14-01-2026

TAREA 4.2 - Informes interactivos con Datapane

Nombre y Apellidos: Cristina
Sandoval Laborde

Curso: 2º DAM

Asignatura: Desarrollo de interfaces

Índice

Ejercicio 1. Filtrado y análisis de ventas	1
1.1. Indica qué filtros y ordenaciones has utilizado.....	1
1.2. Explica brevemente por qué el uso de una tabla interactiva es adecuado para este tipo de consultas.	2
1.3. Indica qué perfil de usuario (dirección, responsable comercial, analista...) se beneficiaría más de este tipo de informe.	3
Ejercicio 2. Diseño de un informe de ventas.....	3
1.El logotipo de la empresa.	3
2.El título del informe: <i>Informe de ventas</i>.....	4
3.Una sección inicial de resumen ejecutivo, que incluya:	5
4.El importe total de ventas del año 2021 y su comparación con el total del año anterior.....	6
5.Una tabla interactiva (DataTable) con los datos del informe, que permita filtrar y ordenar.	7
6.La posibilidad de descargar el fichero con los datos del informe.....	7
•Indica qué región ha tenido mayores ventas en el año 2021.....	7
•Indica qué tipo de producto ha generado más ventas en el conjunto de todos los años.	8
Webgrafía	8

Ejercicio 1. Filtrado y análisis de ventas

Ejercicio 1

Informe de Ventas

Ventas Julio > 500 uds

2

Top: (Noviembre)

5062 €

Top Agosto: Stephanie Cummings

504 Uds

Reporte de Ventas			
Nombre	Mes	Unidades	Importe (€)
1. Renee Martin	Enero	242	4380
2. Angelica Armstrong	Enero	270	2213
3. Chadwick Mosley	Enero	325	5675
4. Boris Vazquez	Enero	343	4298
5. Halee Combs	Enero	431	2636
6. Kirsten Whittfield	Enero	581	5085
7. Halee Combs	Febrero	181	2800
8. Angelica Armstrong	Febrero	297	1144
9. Emerson Stephens	Febrero	490	1544
10. Renee Martin	Febrero	531	1025
11. Chadwick Mosley	Martes	500	3064

Run SQL Query Export ▾



1.1. Indica qué filtros y ordenaciones has utilizado.

Apartado 1 (Ventas Julio > 500):

- Filtro: Columna Mes establecida en "Julio".
- Búsqueda: Columna Unidades con condición mayor que (>) 500

```
vendedores_julio = df[(df['Mes'] == 'Julio') & (df['Unidades'] > 500)]
```

Código completo

```
#Los vendedores que en el mes de julio han vendido más de 500 unidades.
vendedores_julio = df[(df['Mes'] == 'Julio') & (df['Unidades'] > 500)]
total_vendedores = len(vendedores_julio)

indicador_julio = dp.BigNumber(
    heading='Ventas Julio > 500 uds',
    value=total_vendedores
)
```

Apartado 2 (Vendedor máximo importe con < 300 unidades):

- Filtro: Columna Unidades con condición menor que (<) 300.
- Filtro: Uso de la función idxmax() sobre la columna Importe (€) para localizar la posición exacta del valor más alto sin necesidad de ordenar la tabla completa.

```

filtro_unidades = df[df['Unidades'] < 300]
max_importe = filtro_unidades['Importe (€)'].idxmax()
max_vendedor = filtro_unidades.loc[max_importe]

```

Código completo

```

filtro_unidades = df[df['Unidades'] < 300]
max_importe = filtro_unidades['Importe (€)'].idxmax()
max_vendedor = filtro_unidades.loc[max_importe]

nombre = max_vendedor['Nombre']
mes = max_vendedor['Mes']
importe = max_vendedor['Importe (€)']

indicador_max = dp.BigNumber(
    heading=f"Top: ({mes})",
    value=f"{importe} €"
)

```

Apartado 3 (Letra S, máximo unidades en Agosto):

- Filtro: Columna Nombre filtrada por la letra inicial "S".
- Filtro: Columna Mes establecida en "Agosto".
- Búsqueda: Uso de la función idxmax() sobre la columna Unidades para localizar directamente el índice de la fila con el mayor volumen de ventas en ese subconjunto de datos.

```

vendedores_S_agosto = df[(df['Nombre'].str.startswith('S')) & (df['Mes'] == 'Agosto')]
idx_max_S = vendedores_S_agosto['Unidades'].idxmax()
vendedor_S_top = vendedores_S_agosto.loc[idx_max_S]

```

Código completo

```

vendedores_S_agosto = df[(df['Nombre'].str.startswith('S')) & (df['Mes'] == 'Agosto')]
idx_max_S = vendedores_S_agosto['Unidades'].idxmax()
vendedor_S_top = vendedores_S_agosto.loc[idx_max_S]
|
nombre_S = vendedor_S_top['Nombre']
unidades_S = vendedor_S_top['Unidades']

indicador_S = dp.BigNumber(
    heading=f"Top Agosto: {nombre_S}",
    value=f"{unidades_S} Uds"
)

```

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

El uso del componente DataTable es adecuado para este tipo de consultas porque:

- Capacidad de filtrado: Permite mostrar únicamente los datos que cumplen con unas condiciones específicas.
- Uso de múltiples criterios: se pueden aplicar varios filtros a la vez.
- Fácil comparación: Al poder filtrar y ordenar se puede comparar fácilmente valores de diferentes filas.
- Información detallada: además de filtrar, puedes obtener toda la información que necesites.

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

Perfil comercial, porque puede ver el rendimiento en tiempo real, hacer comparaciones para ver quien es el mejor vendedor aplicando filtros, como por semanas o meses, y también hacer comparaciones y ver rápido y fácil si las ventas aumentan o disminuyen mes a mes o por el filtro que quiera por si hay algo que mejorar.

Ejercicio 2. Diseño de un informe de ventas

1. El logotipo de la empresa.

```
# 1. Logotipo de la empresa
imagen_logo = dp.Media(file='ejercicio2.png')
```



Informe de ventas

Resumen Ejecutivo

Total Ventas Acumuladas 122,410.00 €	Año con más ventas 2019
--	-----------------------------------

Datos clave relevantes para la dirección porque permiten evaluar el crecimiento histórico y ayudan a identificar los períodos de mayor éxito comercial para planificar futuras inversiones.

Comparativa anual 2021

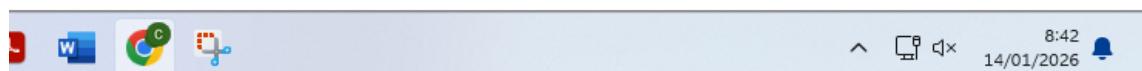
Ventas Año 2021 23,227.00 €	-€11.00 €
---------------------------------------	-----------

Tabla detallada de ventas

Año	Región	Tipo de producto	Ventas
2017	Oeste	Tecnología	1055
2017	Norte	Moda	2607
2017	Sur	Moda	1824
2017	Oste	Moda	1207
2017	Oeste	Moda	1258
2017	Norte	Alimentación	2130
2017	Sur	Alimentación	1872
2017	Oste	Alimentación	1956
2017	Oeste	Alimentación	1339
2018	Norte	Tecnología	2809
2018	Sur	Tecnología	2479

Puedes descargar el fichero con los datos desde aquí:

[D:\U05_A02_PP_E_01.csv](#) Download



2. El título del informe: *Informe de ventas*.

```
# 2. Título del informe
titulo_barra = dp.HTML(
    '<p style="font-size:30px; text-align:center; color:#ffffff; background-color:#4d4d4d;">Informe de ventas</p>')
)
```

Informe de ventas

Resumen Ejecutivo

Total Ventas Acumuladas
122,410.00 €

Año con más ventas
2019

Datos de los son relevantes para la dirección porque permiten evaluar el crecimiento histórico y ayudan a identificar los períodos de mayor éxito comercial para planificar futuras inversiones.

3. Una sección inicial de resumen ejecutivo, que incluya:

- El total de ventas acumuladas de todos los años.

No he utilizado ningún filtro, he procesado los registros del fichero sin restricciones.

Cálculo: Se aplica la función sum() sobre la columna **Ventas** de todo el DataFrame para obtener los 122.410 €.

- El año con mayor volumen de ventas.

Ordenación/Agrupación: Se agrupan los datos por la columna Año y se aplica la función sum()

Búsqueda: Se utiliza la función idxmax() para localizar el año con la cifra más alta, que resulta ser **2019**.

- Un breve texto (2–3 líneas) justificando por qué estos datos son relevantes para la dirección.

```
# --- Cálculos para el Resumen Ejecutivo ---
total_acumulado = df['Ventas'].sum()
# Encontrar el año con más ventas
ventas_por_anio = df.groupby('Año')['Ventas'].sum()
anio_top = ventas_por_anio.idxmax()
```

```
# 3. Resumen ejecutivo
indicador_total = dp.BigNumber([
    heading='Total Ventas Acumuladas',
    value=f'{total_ventas:.2f} €'
])
indicador_anio = dp.BigNumber(
    heading='Año con más ventas',
    value=anio_top
)
texto_justificacion = dp.Text(
    "Estos datos son importantes para el equipo directivo porque permiten evaluar el crecimiento histórico en tiempo real "
    "y ayudan a identificar los períodos de mayor éxito comercial para planificar las acciones futuras."
)
```

Informe de ventas

Resumen Ejecutivo

Total Ventas Acumuladas

122,410.00 €

Año con más ventas

2019

Estos datos son relevantes para la dirección porque permiten evaluar el crecimiento histórico y ayudan a identificar los períodos de mayor éxito comercial para planificar futuras inversiones.

4. El importe total de ventas del año 2021 y su comparación con el total del año anterior.

```
# --- Cálculos para la comparativa 2021 vs 2020 ---
ventas_2021 = df[df['Año'] == 2021]['Ventas'].sum()
ventas_2020 = df[df['Año'] == 2020]['Ventas'].sum()
diferencia_ventas = ventas_2021 - ventas_2020
es_positivo = ventas_2021 > ventas_2020
```

```
# 4. Ventas 2021 y comparación
indicador_2021_total = dp.BigNumber(
    heading='Importe Total Ventas 2021',
    value=f"{ventas_2021:.2f} €"
)
indicador_comparativa = dp.BigNumber(
    heading='Evolución 2021 vs 2020',
    value=f"{ventas_2021:.2f} €",
    change=f"{diferencia_ventas:.2f} €",
    is_upward_change=es_positivo
)
```

Análisis Ventas 2021

Importe Total Ventas 2021

23,227.00 €

Evolución 2021 vs 2020

23,227.00 €

-411.00 €

Filtro: Columna **Año** establecida en el valor **2021**.

Cálculo: Se utiliza la función de agregación sum() sobre la columna **Ventas** del subconjunto filtrado para obtener el valor destacado (23.227,00\$ €).

Comparativa: Se aplica un segundo filtrado en la columna **Año** con el valor **2020** para obtener su suma total (\$23.638,00\$ €). La diferencia resultante (-411,00) se asigna al parámetro **change** del componente BigNumber .

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

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

The screenshot shows a Jupyter Notebook cell with the code `tabla_ventas = dp.DataTable(df)`. Below the code is a rendered visualization of the data as a table titled "Tabla Interactiva". The table has four columns: Año, Región, Tipo de producto, and Ventas. The data consists of 10 rows, each with a row index (1-10), year (2017-2021), region (Norte, Sur, Este, Oeste), product type (Tecnología, Moda, Alimentación), and sales values (e.g., 1980, 2642, 1594, etc.). The table includes standard data table controls like "Run SQL Query" and "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
	2017	Este	Alimentación	1966
	2021	Sur	Moda	2898
	2021	Oeste	Alimentación	2729
	2021	Oeste	Tecnología	2334
	2021	Este	Moda	2311
	2021	Norte	Alimentación	1974
	2021	Oeste	Moda	1956
	2021	Sur	Alimentación	1882
	2021	Este	Tecnología	1760
	2021	Norte	Moda	1707
	2021	Norte	Tecnología	1307
	2021	Sur	Tecnología	1297

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

```
fichero_adjunto = dp.Attachment(file=fichero_csv)
```

Descargar datos originales:

[DI_U05_A02_PP_E_01.csv](#) Download

Utilizando la tabla interactiva del informe:

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

Tabla Interactiva

The screenshot shows a Jupyter Notebook cell with the code `tabla_ventas = dp.DataTable(df)`. Below the code is a rendered visualization of the data as a table titled "Tabla Interactiva". The table has four columns: Año, Región, Tipo de producto, and Ventas. The data consists of 10 rows, each with a row index (1-10), year (2017-2021), region (Norte, Sur, Este, Oeste), product type (Tecnología, Moda, Alimentación), and sales values (e.g., 1980, 2642, 1594, etc.). The first row for the year 2021 (row 11) is highlighted with a red box. The table includes standard data table controls like "Run SQL Query" and "Export".

	Año	Región	Tipo de producto	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	1297

Año 2021, Región sur, tipo de producto moda y ventas 2898

He aplicado un filtro en la columna **Año** seleccionando específicamente el valor **2021** para aislar los datos de ese periodo.

He realizado una **ordenación descendente** (de mayor a menor) en la columna **Ventas**. Al hacerlo, el registro de la región Sur aparece en la primera posición de la tabla.

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

	Año	Región	Tipo de producto	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	

Descargar datos originales:

Tipo de producto: Moda

Para este análisis no he aplicado ningún filtro por año, permitiendo que la tabla mostrara el conjunto completo de datos.

He aplicado una **ordenación descendente** en la columna **Ventas**.

Webgrafía

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