Proyecto final: el dashboard de ventas 2025 de PyShop

Durante el curso cada lección agregó una pieza: líneas, barras, ejes honestos, colores, dispersión, donut, pandas y exportación. En este proyecto se unen todas en un solo producto: el **tablero de ventas 2025 de PyShop**, una figura de una página que la dirección pueda leer en un minuto, exportada a PNG y PDF, y un informe breve con tres hallazgos. Se usan Python 3.11 o superior, matplotlib y pandas. El notebook `proyecto-dashboard` contiene una solución de referencia completa: consúltalo al terminar, no antes.

Contexto

La dirección de PyShop prepara el plan de 2026 y necesita ver el año anterior en una sola hoja: cuánto se vendió, cuándo, qué y dónde, y por qué canal. No quiere veinte gráficos sueltos, sino un tablero con indicadores clave arriba y paneles que respondan una pregunta cada uno.

Datos

La función `generar_ventas(n=6000, semilla=42)` produce los 6000 pedidos de 2025, siempre iguales. Está en la primera celda de código de cualquier notebook del curso; cópiala al inicio del tuyo. Columnas: `fecha`, `categoria`, `region`, `canal`, `precio`, `unidades`, `calificacion` y `total`.

```
import matplotlib.pyplot as plt

ventas = generar_ventas()
print(len(ventas))                      # 6000
print(round(ventas["total"].sum(), 2))  # 10932695.87
```

Requisitos mínimos

1. **Una sola figura** creada con `plt.subplot_mosaic` o `plt.subplots` y `layout="constrained"`, con un título general (`fig.suptitle`).
2. **Fila de KPIs:** ingreso total, pedidos, ticket medio y calificación media, como tarjetas (`ax.axis("off")` y `ax.text`).
3. **Al menos cinco paneles de datos**, cada uno respondiendo una pregunta distinta. Deben aparecer al menos: una tendencia (línea mensual), un ranking

(barras ordenadas), una composición (donut o barras apiladas al 100 %) y una relación (dispersión).
4. **Ejes honestos:** barras desde cero, nombres de mes en el eje x, unidades en títulos o etiquetas y formato legible de los montos (por ejemplo «1.5 M» con un formateador).
5. **Estilo consistente:** una paleta de pocos colores, un color de resalte para lo importante y gris para el contexto; sin bordes superior y derecho donde no aporten.
6. **Una función por panel** que reciba un `Axes` y un DataFrame, y una función `dashboard(df, titulo)` que arme la figura completa.
7. **Exportación:** el tablero en PNG a 300 dpi y en PDF. Comprueba el tamaño del PNG en píxeles.

Cifras esperadas

Tu tablero debe mostrar estos valores:

- KPIs: ingreso 10.93 M MXN, 6000 pedidos, ticket medio 1,822.12 MXN y calificación media 4.14.
- Ingreso mensual: mínimo en febrero (0.61 M) y máximo en diciembre (1.50 M).
- Trimestres: 2.27, 2.38, 2.55 y 3.73 M; el cuarto trimestre es el 34.1 % del año.
- Categorías: Electrónica 6.10, Hogar 1.81, Deportes 1.40, Ropa 1.26 y Libros 0.36 M.
- Regiones: Centro 4.22, Norte 2.51, Occidente 2.46 y Sur 1.74 M.
- Canales: Web 44.2 %, App 37.3 % y Marketplace 18.5 %.
- Correlación entre pedidos e ingreso diarios: 0.787.

Si una cifra no coincide, el error casi siempre está en la agrupación: revisa la columna del `groupby` y si dividiste entre un millón.

Pistas de implementación

En `subplot_mosaic` cada fila es una lista de nombres; un nombre repetido ocupa varias celdas. El resultado es un diccionario de paneles:

```
MOSAICO = [
    ["k1", "k2", "k3", "k4"],
    ["mes", "mes", "mes", "trim"],
    ["cat", "cat", "reg", "canal"],
    ["disp", "disp", "disp", "disp"],
]
fig, axd = plt.subplot_mosaic(
```

```
    MOSAICO, figsize=(16, 13), layout="constrained",
    height_ratios=[0.45, 1, 1, 0.9])
axd["mes"].set_title("Ingreso mensual", loc="left")
```

Un formateador evita dividir a mano en cada panel:

```
from matplotlib.ticker import FuncFormatter

def millones(v, pos):
    return f"{v / 1e6:.1f} M"

ax.yaxis.set_major_formatter(FuncFormatter(millones))
```

Para la exportación, 16 × 13 pulgadas a 300 dpi dan 4800 × 3900 píxeles. Si usas `bbox_inches="tight"`, el tamaño cambia porque se recortan los márgenes.

Informe breve

Acompaña el tablero con un texto de media página para la dirección: tres hallazgos, cada uno con una cifra y el panel que lo respalda, y una línea sobre lo que los datos *no* muestran. Ejemplo de formato: «El cuarto trimestre aporta el 34.1 % del ingreso (panel Trimestres)». Ten en cuenta que el día de la semana no tiene patrón en estos datos y que las calificaciones por categoría son casi iguales (4.09 a 4.17): no inventes una explicación.

Extras opcionales

- **Versión por región:** llama a `dashboard` con el DataFrame filtrado de una región y exporta un PNG por región.
- **Estilo oscuro:** genera una versión dentro de `plt.style.context("dark_background")` que conserve la legibilidad.
- **Anotaciones:** señala el máximo mensual con `annotate`.
- **Burbujas:** agrega un panel región × categoría con tamaño proporcional al ingreso.

Entregables

1. **Un notebook** que corra de principio a fin sin errores: datos, cifras de control, funciones de panel, tablero, exportación e informe.
2. **Dos archivos:** `dashboard_2025.png` (300 dpi) y `dashboard_2025.pdf`.

3. **El informe breve** con tres hallazgos, en una celda de texto del notebook o en un documento aparte.

Rúbrica (100 puntos)

- **Datos y cifras (20 puntos):** agrupaciones correctas y cifras esperadas en el tablero.
- **Elección de gráficos (20 puntos):** cada panel responde una pregunta con el tipo adecuado.
- **Ejes y legibilidad (20 puntos):** barras desde cero, etiquetas, unidades y formato de montos.
- **Diseño del tablero (15 puntos):** mosaico ordenado, KPIs arriba, paleta consistente y sin elementos encimados.
- **Código reutilizable (10 puntos):** funciones por panel y `dashboard(df, titulo)`.
- **Exportación (5 puntos):** PNG a 300 dpi y PDF verificados.
- **Informe (10 puntos):** tres hallazgos con cifras y una limitación de los datos.

Los extras suman hasta 10 puntos adicionales y no sustituyen a los requisitos mínimos.