Proyecto final: la bandeja de soporte de PyShop en una tabla

Durante el curso cada herramienta de regex resolvió una parte del caos de la bandeja de soporte: pedidos escritos de tres formas, fechas en tres formatos, teléfonos en cuatro y montos con o sin signo de pesos. En este proyecto se unen todas en un solo proceso: de 600 tickets escritos a mano a una tabla limpia, una versión anonimizada que se puede compartir y un reporte con cifras. Se usa Python 3.11 o superior, el módulo `re`, `collections.Counter` y pandas. El notebook `proyecto-tickets` contiene una solución de referencia completa: consúltalo al terminar, no antes.

Contexto

El equipo de soporte de PyShop quiere saber de qué se quejan los clientes, cuánto dinero está en juego en cada tipo de queja y en qué meses se concentran. Hoy esa información está atrapada en texto libre. Cada ticket tiene tres líneas: el identificador, el remitente y el mensaje:

```
Ticket T-0002
De: María Gómez <maria.gomez90@ejemplo.org>
Hola, la guía del pedido PS-46421 no se actualiza.
Pagué 4491.00 MXN con efectivo el 2026-06-25.
Mi teléfono es 8116155940.
```

En la bandeja real el mensaje va en una sola línea; aquí se partió para que quepa.

Datos

La función `generar_tickets(n=600, semilla=42)` produce la bandeja, siempre igual. Está en la primera celda de código de cualquier notebook del curso, junto con `generar_log` y la lista `MESES`; cópiala al inicio de tu notebook. Para clasificar el tema se usan estas palabras clave, revisadas en orden (gana el primer tema que coincide):

```
TEMAS = {
    "envio": r"no ha llegado|retraso|guía",
    "devolucion": r"devolver|reembolso|cambiar",
    "pago": r"cobraron|rechazado|reflejado",
    "danado": r"dañado|roto|faltan",
    "factura": r"factura",
}
```

```
tickets = generar_tickets()
print(len(tickets))  # 600
```

Requisitos mínimos

1. **Patrones con nombre.** Define cada patrón una sola vez con `re.compile`, en un bloque o módulo `patrones.py`, con al menos dos pruebas `assert` por patrón: una que debe coincidir y otra que no.
2. **Tabla limpia.** Un DataFrame con una fila por ticket y las columnas `ticket`, `cliente`, `dominio`, `pedido` (normalizado a `PS-ddddd`), `fecha` (ISO `aaaa-mm-dd`), `monto` (`float`), `metodo`, `tema` y `telefono` (10 dígitos). Si un ticket no trae el dato, la celda queda vacía (`None` o `NaN`), nunca con un valor inventado.
3. **Fechas.** Convierte los tres formatos (`2026-06-25`, `28/02/2026` y «24 de febrero de 2026») a ISO con `fullmatch` y grupos.
4. **Montos.** Extrae el monto que sigue a «Pagué» en sus tres formatos y conviértelo a número.
5. **Versión anonimizada.** Reemplaza cada correo por `***@dominio` y cada teléfono por `** **** ****`, y demuestra con una búsqueda que no queda ninguno.
6. **Calidad de datos.** Cuenta los tickets cuyo pedido no viene en el formato oficial `PS-ddddd` y los pedidos repetidos.
7. **Reporte.** Tickets por tema y por mes, monto total por tema, métodos de pago y las palabras más frecuentes del cuerpo (4 letras o más).

Cifras esperadas

Tu solución debe reproducir estos valores:

- 600 tickets, 600 pedidos normalizados y 597 distintos (3 repetidos); 131 tickets traen el pedido con minúsculas o con espacio.
- 488 tickets con fecha, todas convertidas a ISO (0 sin convertir). Por mes: enero 44, febrero 55, marzo 51, abril 43, mayo 39, junio 51, julio 58, agosto 53, septiembre 51 y octubre 43.
- 488 montos, que suman 1,211,689.39 MXN; media 2,482.97 y máximo 4,984.50.
- Métodos de pago: efectivo 151, tarjeta 118, PayPal 115 y transferencia 104.
- 369 tickets con teléfono. Dominios: correo.com 167, ejemplo.org 156, mail.net 139 y pyshop.mx 138.
- Temas: devolución 131, dañado 129, envío 126, factura 117 y pago 97; ningún ticket sin tema.
- Monto por tema: dañado 266,843.12, envío 262,079.44, factura 259,447.61, devolución 255,815.33 y pago 167,503.89.
- Palabras más frecuentes: hola 600, pedido 600, pagué 488, teléfono 369 y efectivo 151.

- Anonimizada: 0 correos y 0 teléfonos restantes.

Si una cifra no coincide, el error casi siempre está en un formato que el patrón no cubre: imprime los tickets en los que la columna quedó vacía.

Pistas de implementación

pandas aplica regex a una columna completa con `.str.extract`, que devuelve una columna por grupo; los grupos con nombre se convierten en el nombre de la columna:

```
import pandas as pd

df = pd.DataFrame({"texto": tickets})
num = df["texto"].str.extract(
    r"(?i)\bps[- ]?(\d{5})\b", expand=False)
df["pedido"] = "PS-" + num
print(df["pedido"].nunique())  # 597
```

Para las fechas y los montos conviene una función de Python por formato y `.map`: un patrón gigante con todas las alternativas es difícil de probar.

Extras opcionales

- **Log del servidor.** Con `generar_log()`, extrae los campos con un patrón de grupos con nombre: las 2000 líneas deben coincidir. Estados esperados: 200 con 1623, 201 con 34, 301 con 93, 404 con 192 y 500 con 58; 45 peticiones tardan más de 1 s.
- **Cruce con el log.** Normaliza las rutas a `/api/pedidos/{id}` (232 peticiones GET) y explica por qué no se pueden cruzar con los pedidos de la bandeja: los identificadores del log van de 1 a 500 y los pedidos tienen cinco dígitos.
- **Validación de calendario.** Comprueba cada fecha ISO con `datetime.date.fromisoformat`.
- **Gráfica.** Tickets por mes y monto por tema con matplotlib.

Entregables

1. **Un notebook** que corra de principio a fin sin errores, con la estructura: datos, patrones y pruebas, tabla limpia, anonimización, calidad de datos, reporte y conclusiones.
2. **Dos archivos CSV:** la tabla limpia y la versión anonimizada.
3. **Un resumen** de media página para el equipo de soporte: temas principales, monto en juego por tema, meses con más tickets y qué formato de captura

conviene exigir en el formulario.

Rúbrica (100 puntos)

- **Patrones y pruebas (20 puntos):** compilados, con nombre, raw strings y pruebas positivas y negativas.
- **Tabla limpia (25 puntos):** las nueve columnas, valores vacíos donde falta el dato y cifras esperadas.
- **Fechas y montos (15 puntos):** los tres formatos de cada uno convertidos sin pérdidas.
- **Anonimización (15 puntos):** correos y teléfonos reemplazados y verificación de 0 restantes.
- **Reporte (15 puntos):** tablas por tema, mes, monto y método, y calidad de datos.
- **Comunicación (10 puntos):** resumen claro, con cifras y recomendaciones concretas.

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