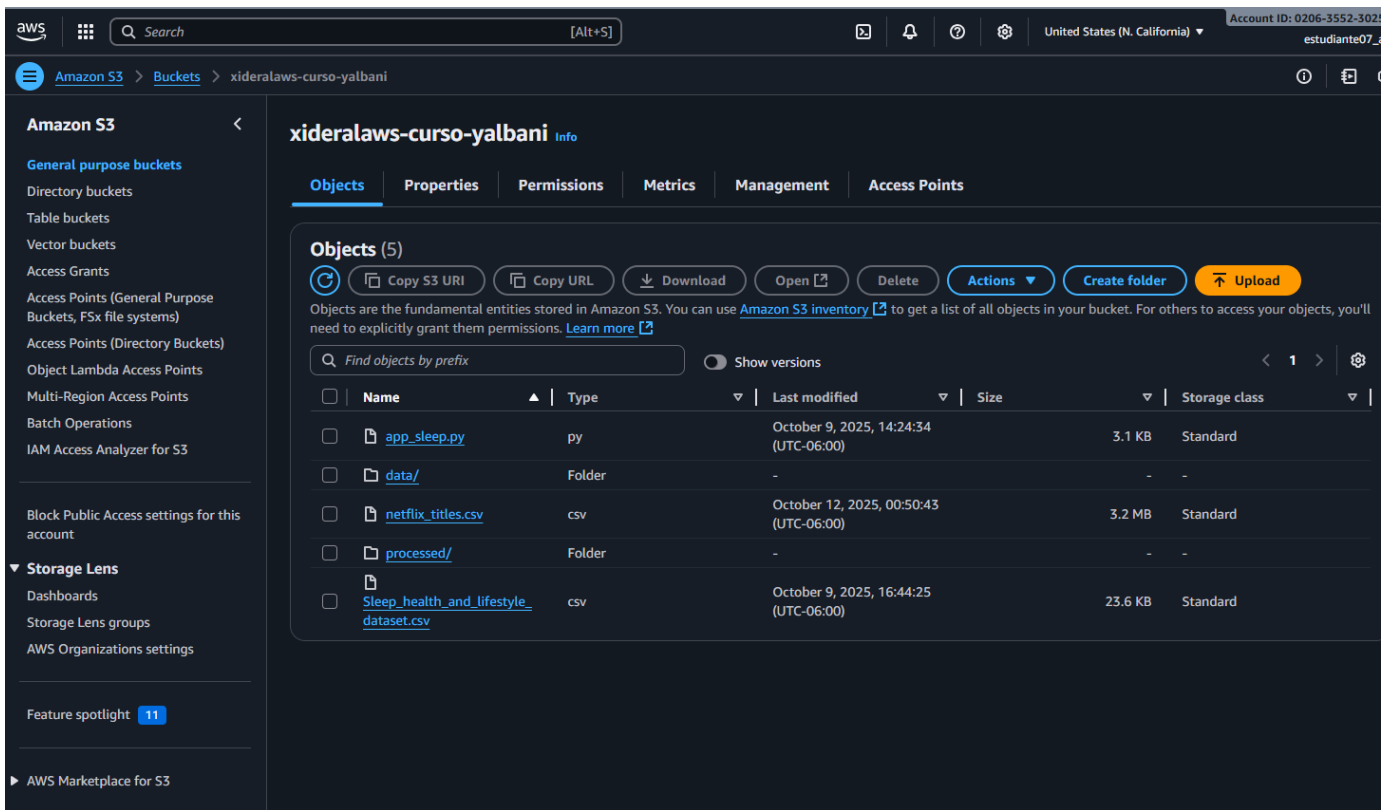
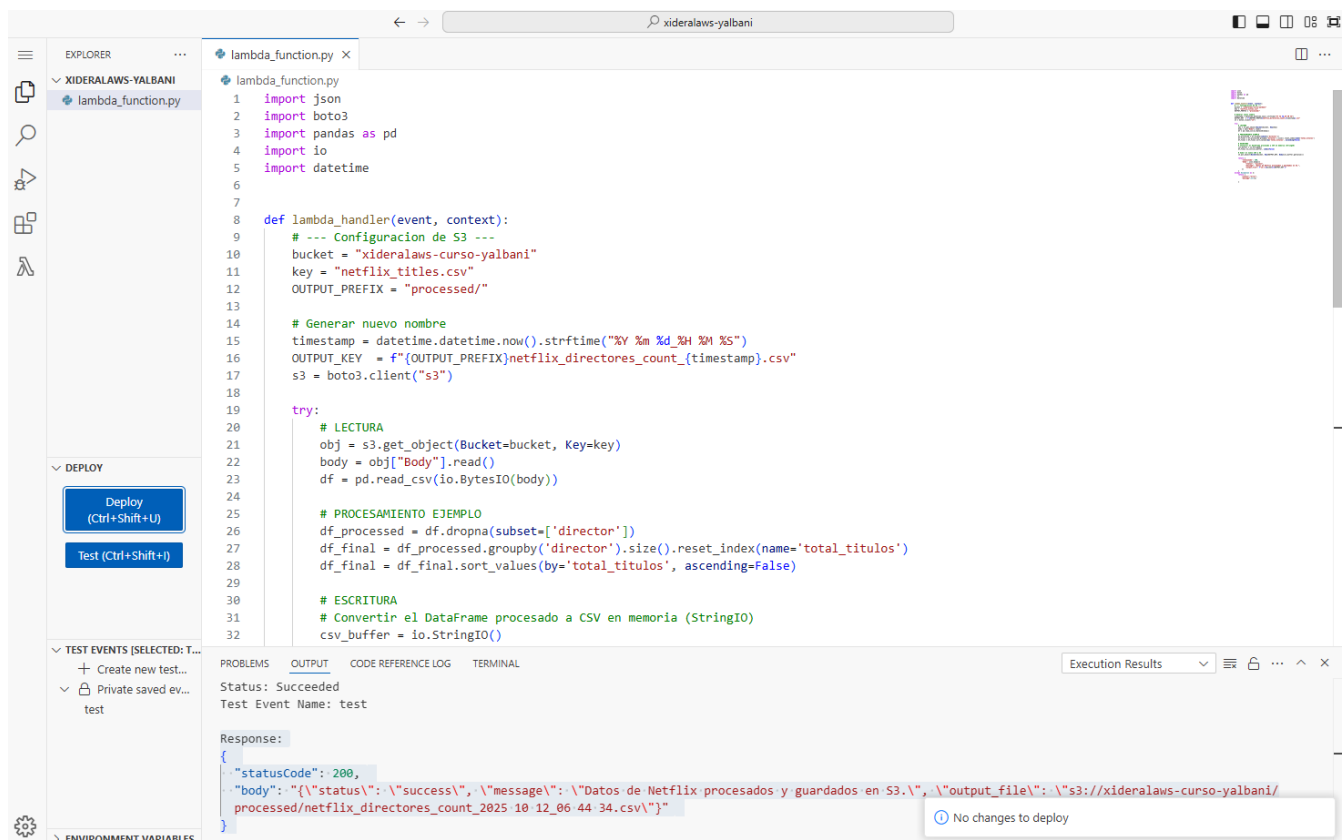


# LAMBDA A STRAMLIT

- Contenido S3



- Funcion LAMBDA. Guarda procesamiento en carpeta processed



- Con el csv guardado, se activa el streamlit

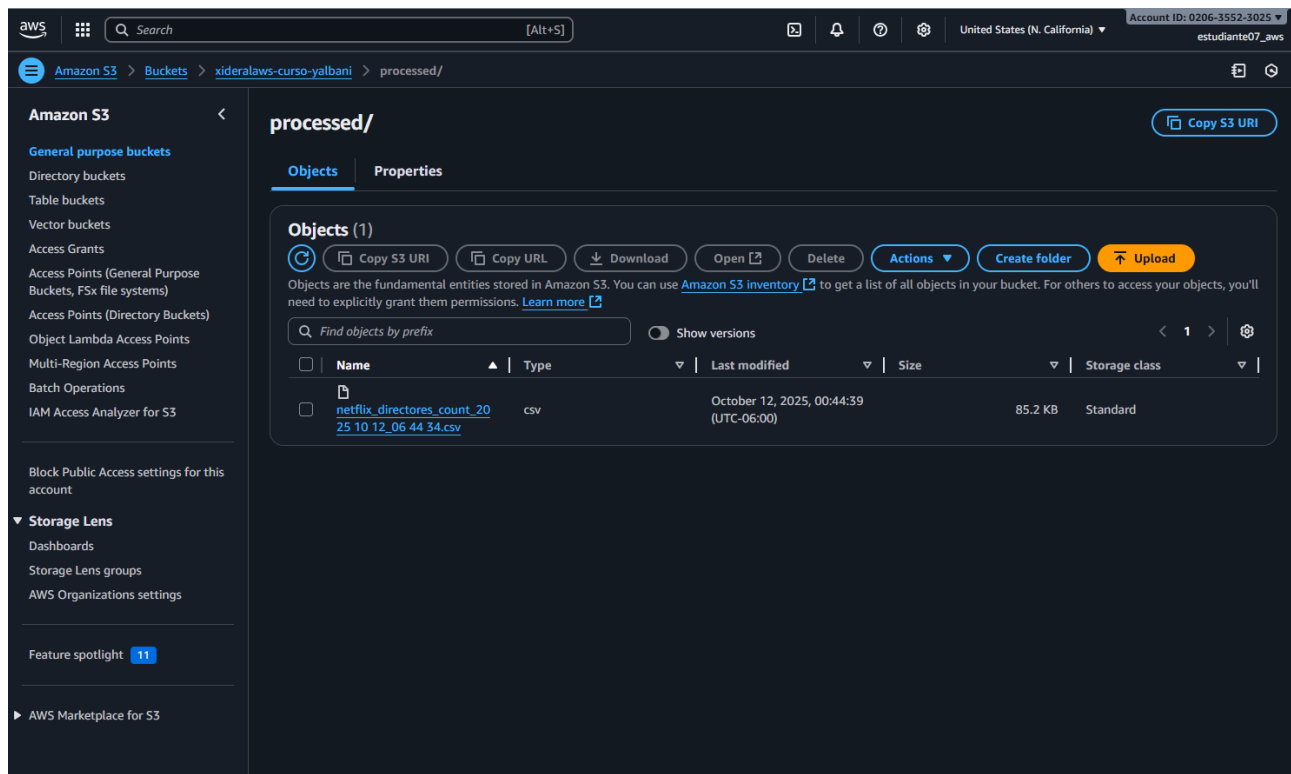
```

jupyter app_lambda.py Last Checkpoint: 6 minutes ago
File Edit View Settings Help

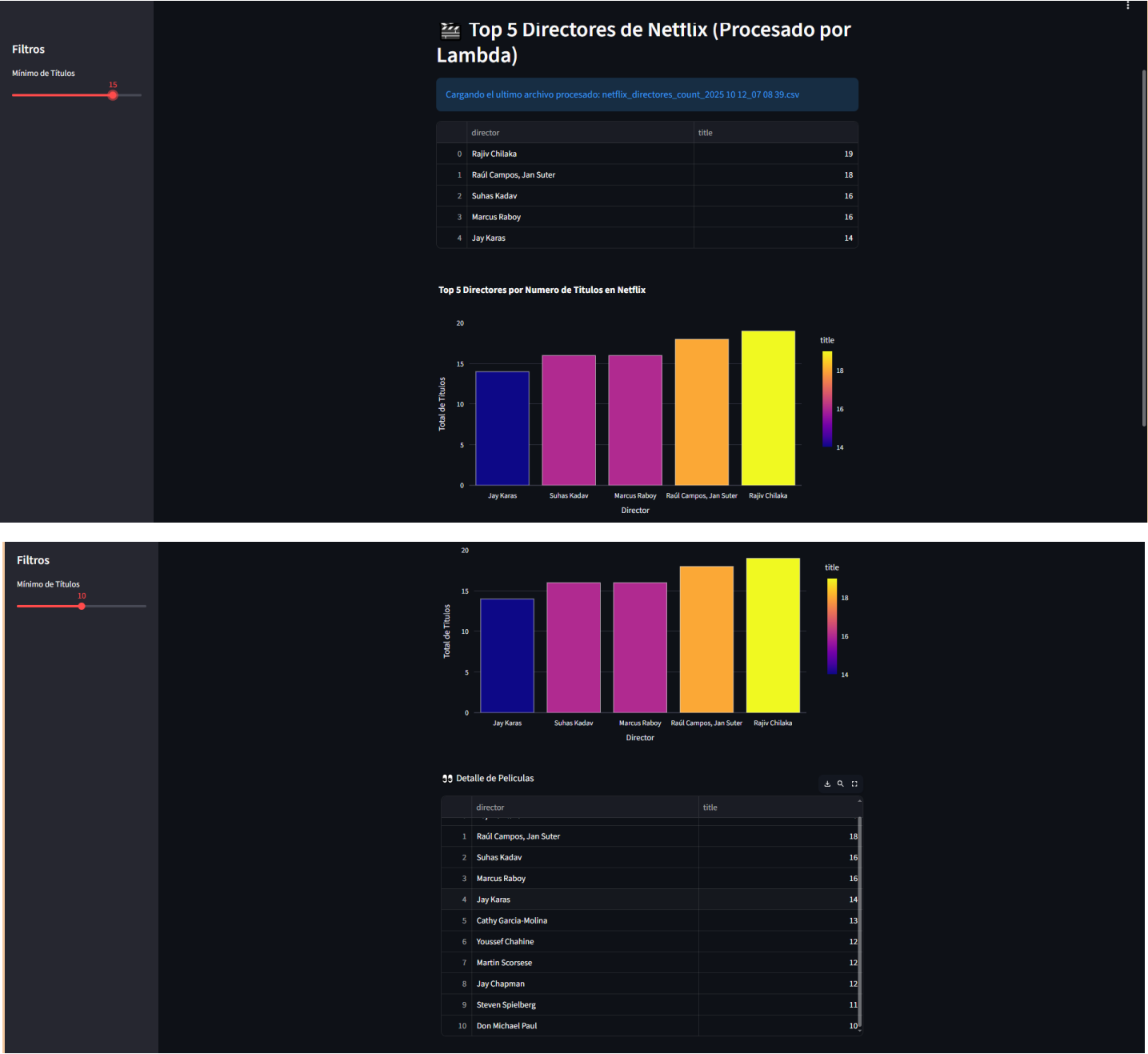
1 import pandas as pd
2 import streamlit as st
3 import boto3
4 import io
5 import requests
6 import json
7 import plotly.express as px
8
9 # --- Carga de Datos Procesados desde S3 ---
10 @st.cache_data
11 def cargar_datos_procesados():
12     s3 = boto3.client("s3")
13     bucket = "xideralaws-curso-yalban1"
14     OUTPUT_PREFIX = "processed/"
15
16     try:
17         # Listar todos los archivos en la carpeta procesada
18         response = s3.list_objects_v2(Bucket=bucket, Prefix=OUTPUT_PREFIX)
19
20         if "Contents" not in response:
21             st.warning("Lambda no ha ejecutado el procesamiento. No se encontraron archivos procesados.")
22             return pd.DataFrame()
23
24         # Encontrar el archivo mas reciente (timestamp mas alto)
25         all_files = response["Contents"]
26         # Filtrar solo archivos CSV si fuera necesario, y ordenar por la ultima modificacion
27         all_files = sorted(all_files, key=lambda x: x['LastModified'], reverse=True)
28
29         # El archivo mas reciente es el primero
30         latest_key = all_files[0]['Key']
31
32         # Leer el archivo mas reciente
33         st.info(f"Cargando el ultimo archivo procesado: {latest_key.split('/')[-1]}")
34         obj = s3.get_object(Bucket=bucket, Key=latest_key)
35         body = obj["Body"].read()
36
37         df_final = pd.read_csv(io.BytesIO(body))
38         return df_final
39
40     except Exception as e:
41         st.error(f"Error al cargar el CSV procesado desde S3: {e}")
42         return pd.DataFrame()
43
44 # --- Streamlit ---
45

```

- Aquí se guarda el csv procesado



- Resultado Streamlit



- Crear un CRON que ejecute diario de Lunes a Viernes a las 13 horas y agregarlo a la lambda

`cron(0 13 ? * MON-FRI *)`

The screenshot displays the AWS Lambda console for the function 'xideralaws-yalbani'. At the top, a green notification bar states: 'The trigger lambda-13-l-v was successfully added to function xideralaws-yalbani. The function is now receiving events from the trigger.' Below this, the 'Function overview' section includes a 'Diagram' tab, a visual representation of the function with its layers, and buttons for '+ Add trigger' and '+ Add destination'. To the right, a 'Description' box shows the function's ARN: 'arn:aws:lambda:us-west-1:020635523025:function:xideralaws-yalbani'. The 'Configuration' tab is active, showing a left-hand menu with options like 'General configuration', 'Triggers', 'Permissions', 'Destinations', 'Function URL', 'Environment variables', and 'Tags'. The main area of the 'Configuration' tab is titled 'Triggers (0)' and contains a search bar and a table with one entry labeled 'Trigger'. At the bottom right of the triggers section, there are buttons for 'Fix errors', 'Edit', 'Delete', and '+ Add trigger'.