

TAREA 5

LAMBDA A STRAMLIT

- Contenido S3

The screenshot shows the Amazon S3 console interface. The left sidebar contains navigation options for 'Amazon S3', including 'General purpose buckets', 'Access Grants', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'IAM Access Analyzer for S3'. The main panel displays the 'xideralaws-curso-yalbani' bucket with 5 objects. The objects are listed in a table with columns for Name, Type, Last modified, Size, and Storage class.

Name	Type	Last modified	Size	Storage class
app_sleep.py	py	October 9, 2025, 14:24:34 (UTC-06:00)	3.1 KB	Standard
data/	Folder	-	-	-
netflix_titles.csv	csv	October 12, 2025, 00:50:43 (UTC-06:00)	3.2 MB	Standard
processed/	Folder	-	-	-
Sleep_health_and_lifestyle_dataset.csv	csv	October 9, 2025, 16:44:25 (UTC-06:00)	23.6 KB	Standard

- Funcion LAMBDA. Guarda procesamiento en carpeta processed

The screenshot shows the AWS Lambda console interface. The left sidebar contains navigation options for 'XIDERALAWS-YALBANI', including 'lambda_function.py'. The main panel displays the code for the 'lambda_function.py' file. The code is a Python function that processes data from a CSV file and uploads it to an S3 bucket. The function is named 'lambda_handler' and takes 'event' and 'context' as arguments. It uses the 'boto3' library to interact with S3. The code includes comments in Spanish describing the configuration and processing steps.

```
1 import json
2 import boto3
3 import pandas as pd
4 import io
5 import datetime
6
7
8 def lambda_handler(event, context):
9     # --- Configuración de S3 ---
10    bucket = "xideralaws-curso-yalbani"
11    key = "netflix_titles.csv"
12    OUTPUT_PREFIX = "processed/"
13
14    # Generar nuevo nombre
15    timestamp = datetime.datetime.now().strftime("%Y %m %d %H %M %S")
16    OUTPUT_KEY = f"{OUTPUT_PREFIX}netflix_directores_count_{timestamp}.csv"
17    s3 = boto3.client("s3")
18
19    try:
20        # LECTURA
21        obj = s3.get_object(Bucket=bucket, Key=key)
22        body = obj["Body"].read()
23        df = pd.read_csv(io.BytesIO(body))
24
25        # PROCESAMIENTO EJEMPLO
26        df_processed = df.dropna(subset=['director'])
27        df_final = df_processed.groupby('director').size().reset_index(name='total_titulos')
28        df_final = df_final.sort_values(by='total_titulos', ascending=False)
29
30        # ESCRITURA
31        # Convertir el DataFrame procesado a CSV en memoria (StringIO)
32        csv_buffer = io.StringIO()
```

The console shows the execution results of the function. The status is 'Succeeded' and the response is a JSON object with a 'statusCode' of 200 and a 'body' containing a success message and the output file path.

```
{
  "statusCode": 200,
  "body": "{\"status\": \"success\", \"message\": \"Datos de Netflix procesados y guardados en S3.\", \"output_file\": \"s3://xideralaws-curso-yalbani/processed/netflix_directores_count_2025 10 12 06 44 34.csv\"}"
}
```

- 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-yalbani"
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

Amazon S3 > Buckets > xideralaws-curso-yalbani > processed/

Amazon S3

General purpose buckets

Directory buckets

Table buckets

Vector buckets

Access Grants

Access Points (General Purpose Buckets, FSx file systems)

Access Points (Directory Buckets)

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Feature spotlight 11

AWS Marketplace for S3

processed/

Objects (1)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix Show versions

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	netflix_directores_count_2025_10_12_06_44_34.csv	csv	October 12, 2025, 00:44:39 (UTC-06:00)	85.2 KB	Standard

- Resultado Streamlit

