

Adaía Montaño

Federico Libertun

Programación Avanzada para grandes volúmenes de datos

Informe Trabajo Práctico Final

Maestría en Ciencia de Datos

26-06-2025

Universidad de San Andrés

Informe de Implementación - TP Final

# 1. Resumen General del Sistema

El sistema desarrollado implementa una pipeline de procesamiento de datos en Airflow, una API en FastAPI, una base de datos en PostgreSQL y un despliegue automatizado utilizando servicios de Google Cloud Platform (GCP). Todo esto está orientado a resolver el problema de recomendaciones de productos para anunciantes en un entorno AdTech.

# 2. Pipeline en Airflow

La pipeline está compuesta por las siguientes tareas:  
- Extracción de logs crudos desde el bucket de Cloud Storage `adtech-tp-data/raw`  
- Filtrado de líneas por advertisers activos utilizando `advertisers.csv`  
- Cálculo del modelo TopCTR: productos con mejor click-through-rate  
- Cálculo del modelo TopProduct: productos más vistos por advertiser  
- Escritura de resultados en las tablas `top\_ctr` y `top\_product` en PostgreSQL

# 3. API en FastAPI

La API la armamos con FastAPI y la desplegada en Google Cloud Run. Se puede acceder desde este link: <https://recom-api-7120503241.us-central1.run.app/docs#/>

La API fue dockerizada utilizando un Dockerfile, y luego subida a un repositorio en Docker Hub bajo el nombre adaiamontano/recom-api. Posteriormente, se desplegó el contenedor desde Cloud Run, permitiendo el acceso público. Esto cumple con los requisitos de contenerización solicitados en el TP.  
  
Tiene los siguientes endpoints implementados:  
- /recommendations/{adv\_id}/{model}: devuelve hasta 20 productos recomendados.  
- /history/{adv\_id}: muestra el historial de productos recomendados de los últimos 7 días.  
- /stats/: devuelve métricas generales como cantidad de anunciantes, productos y registros.  
  
Probamos todos los endpoints desde Swagger y devuelven bien según los datos cargados.

# 4. Base de Datos PostgreSQL

Se utilizaron dos tablas principales:  
- `top\_ctr(advertiser\_id, product\_id, fecha)`  
- `top\_product(advertiser\_id, product\_id, views, fecha)`  
Los resultados diarios calculados por Airflow son cargados en estas tablas. Las consultas sobre la fecha '2025-05-25' verificaron que los datos están almacenados correctamente.

# 5. Validación

Validamos el funcionamiento correcto de la API comparando las respuestas JSON generadas en FastAPI con los archivos `studio\_results\_TOP CTR.csv` y `studio\_results\_TOP Product.csv`. Las coincidencias verifican que la lógica de modelos y la persistencia en base de datos están bien implementadas.

# 6. Dificultades Encontradas

- Ajustes manuales de credenciales y conexión en Cloud SQL  
- Autenticación de Docker para el push de imágenes  
- Configuración de Airflow con LocalExecutor  
- Manejo de rutas absolutas en GCP SSH para descargas de archivos

# 7. Conclusión

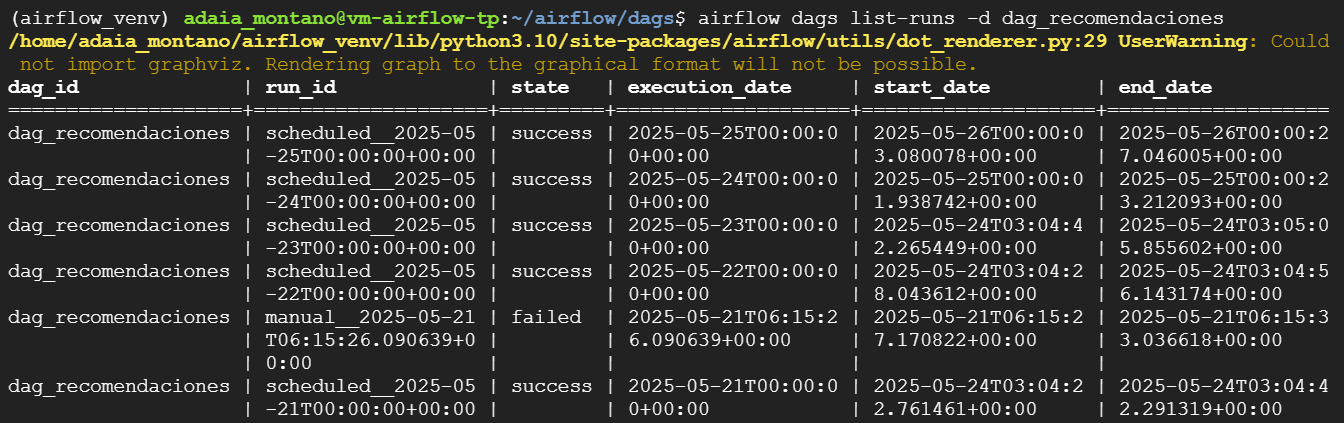
El trabajo cumple todos los requisitos establecidos en el enunciado del TP. El sistema procesa automáticamente los datos en el bucket, genera recomendaciones y las expone mediante una API siendo el despliegue probado y validado adjuntamos las imagenes en el siguiente punto.

# Evidencia de Funcionamiento

Incluimos en el informe técnico la evidencia de la implementación de la pipeline de datos, la API para exponer resultados y todo el despliegue en la nube.

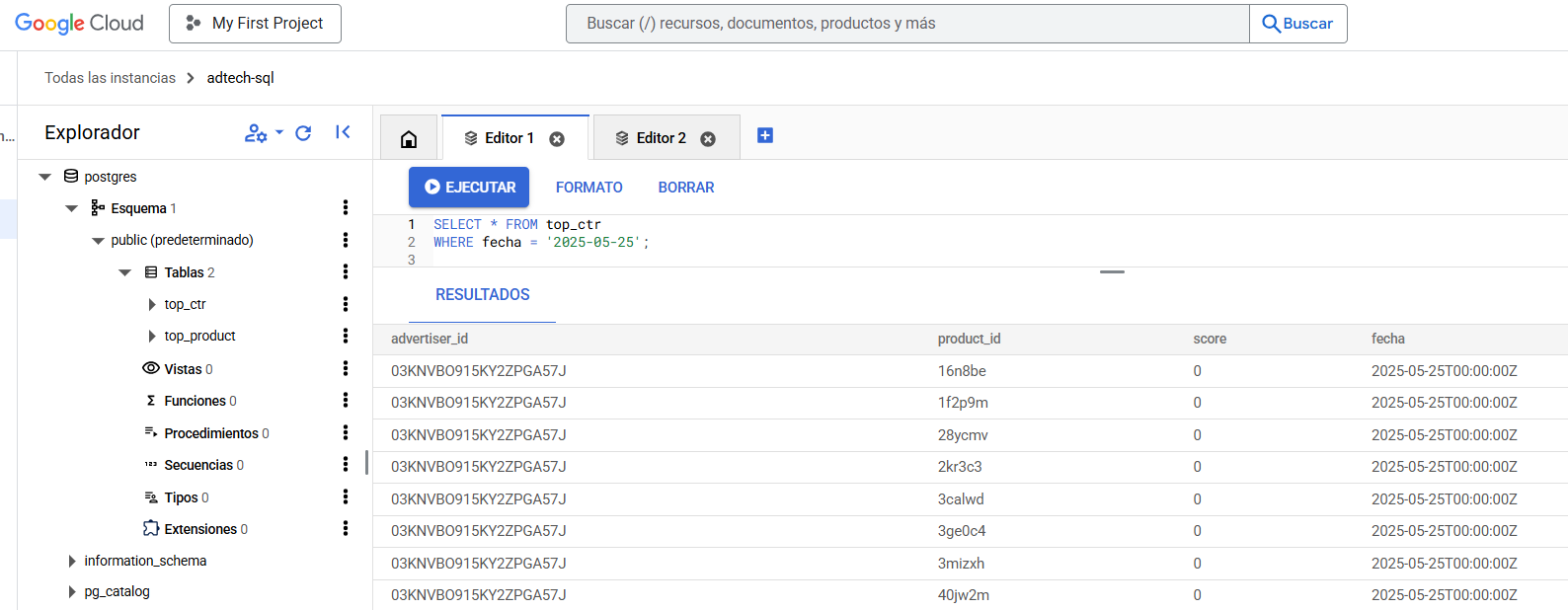
## Ejecución del DAG en Airflow

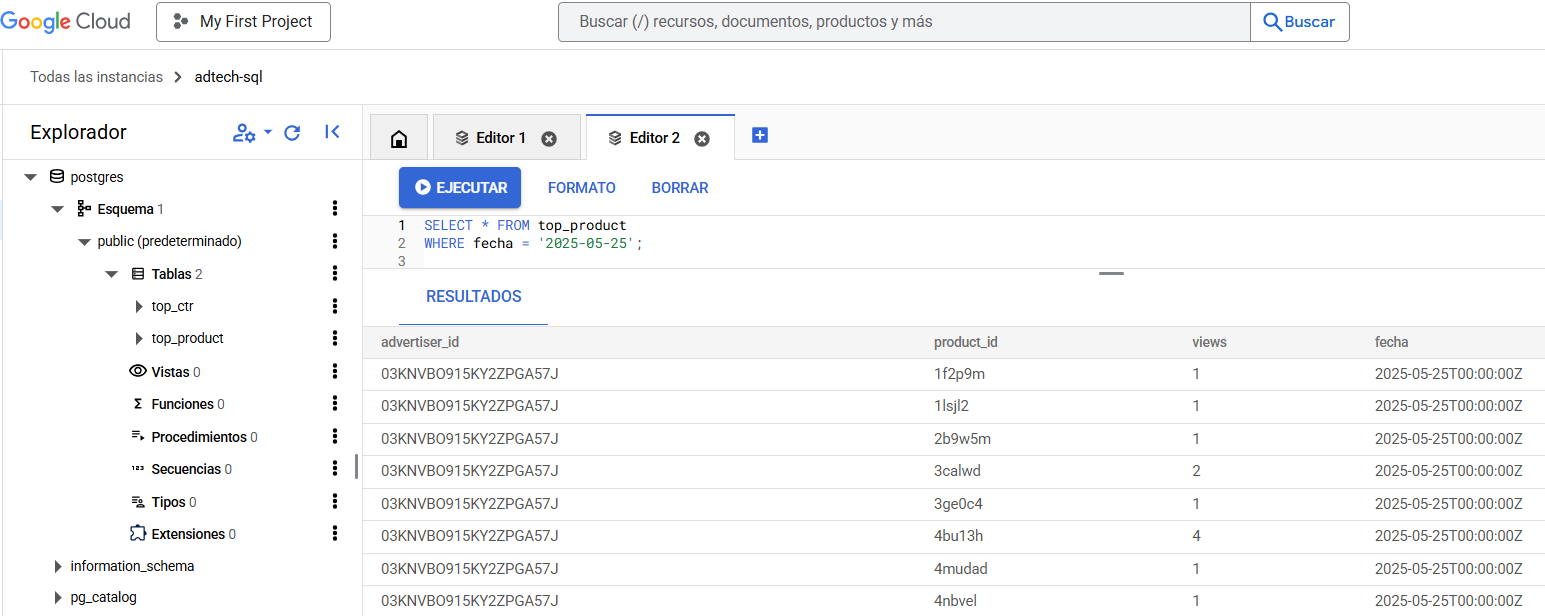
El siguiente registro muestra el historial de ejecuciones del DAG dag\_recomendaciones, ejecutado diariamente mediante Apache Airflow.



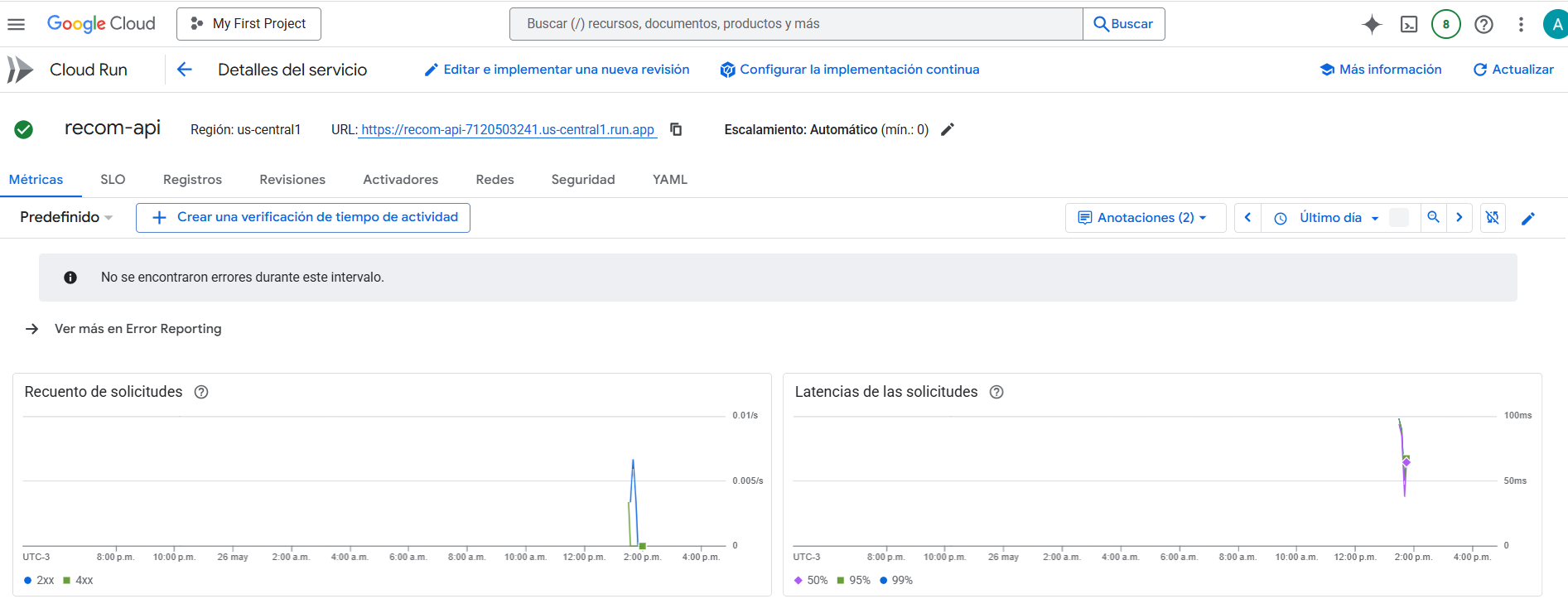
Estas ejecuciones reflejan la estabilidad lograda en la configuración actual y son prueba de que el sistema procesa automáticamente la información cargada en el bucket adtech-tp-data, transformándola y cargándola correctamente en las tablas top\_ctr y top\_product de la base de datos PostgreSQL de Cloud SQL.

## Carga y validación en PostgreSQL

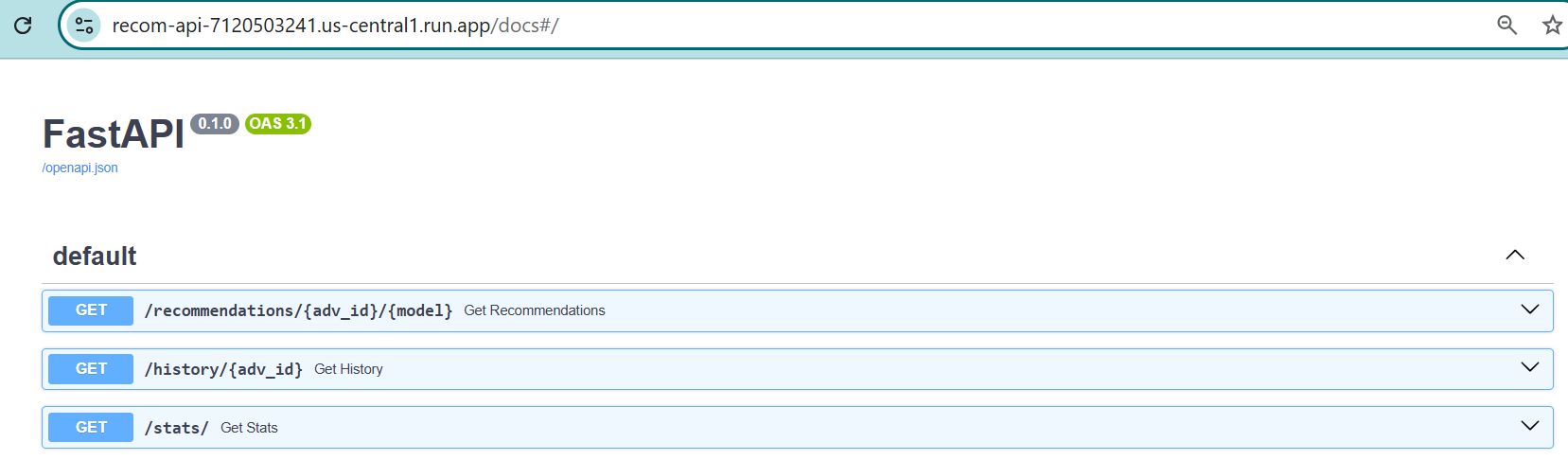




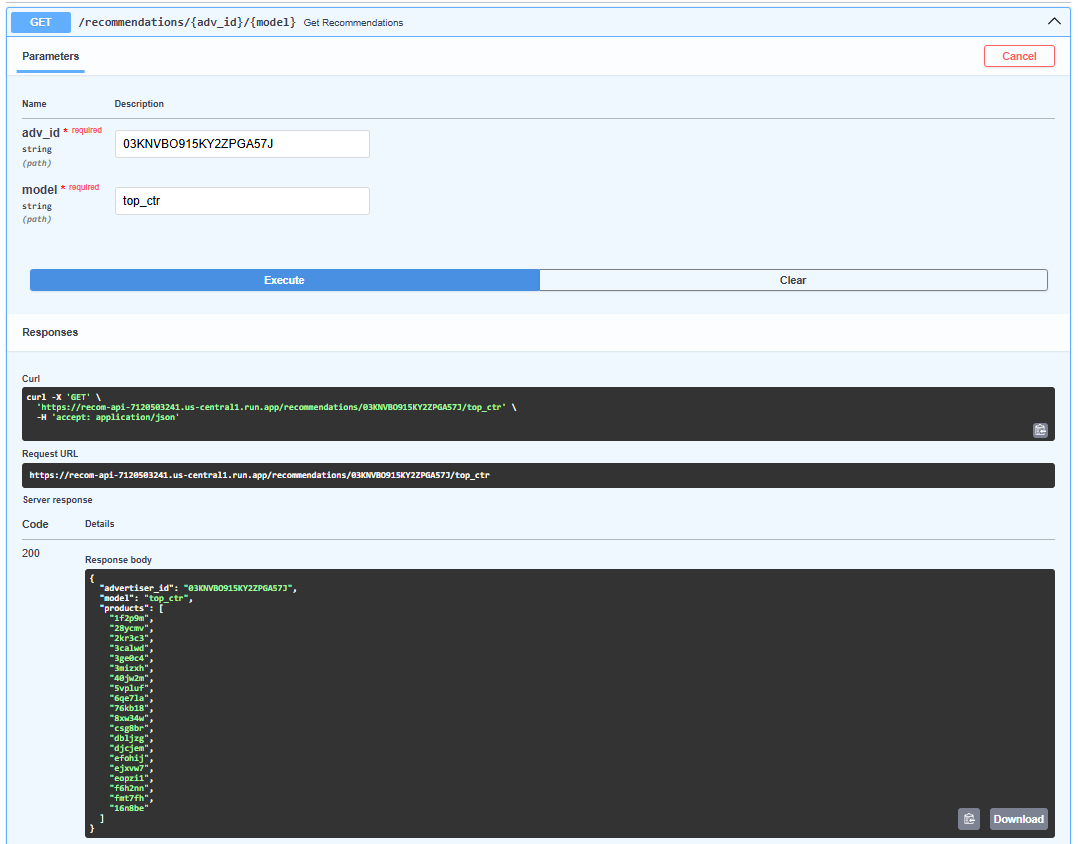
## Despliegue en Cloud Run

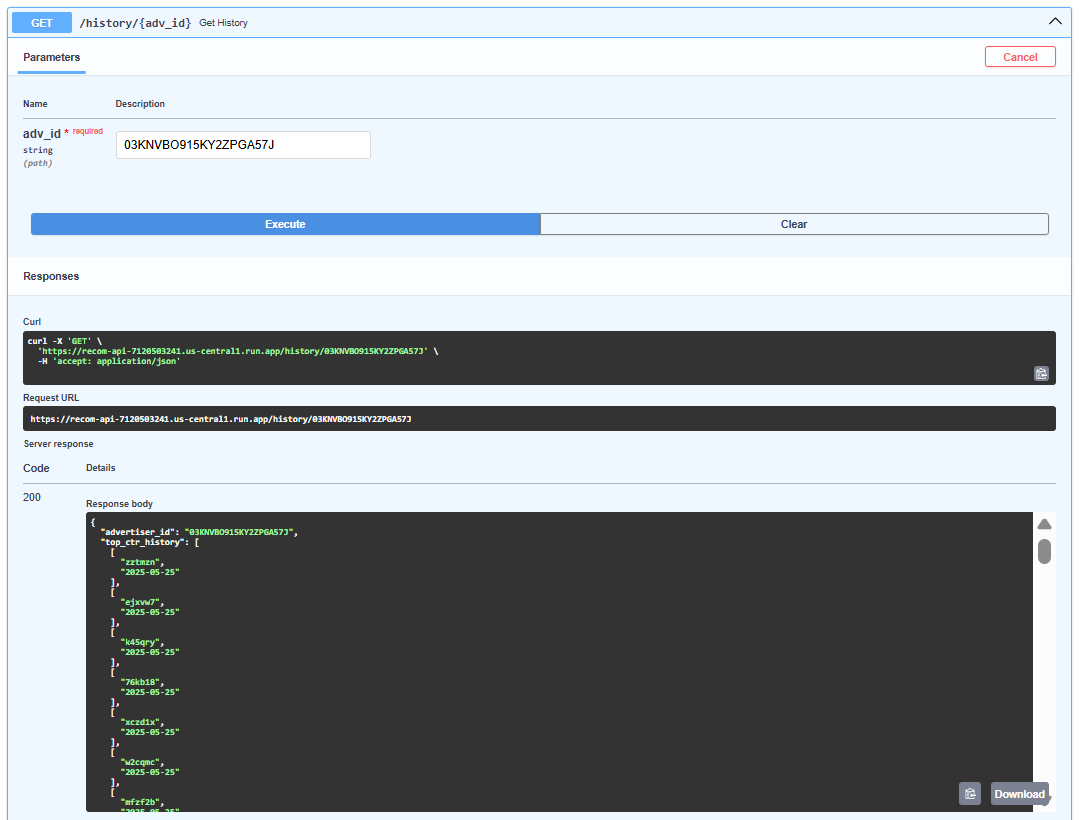


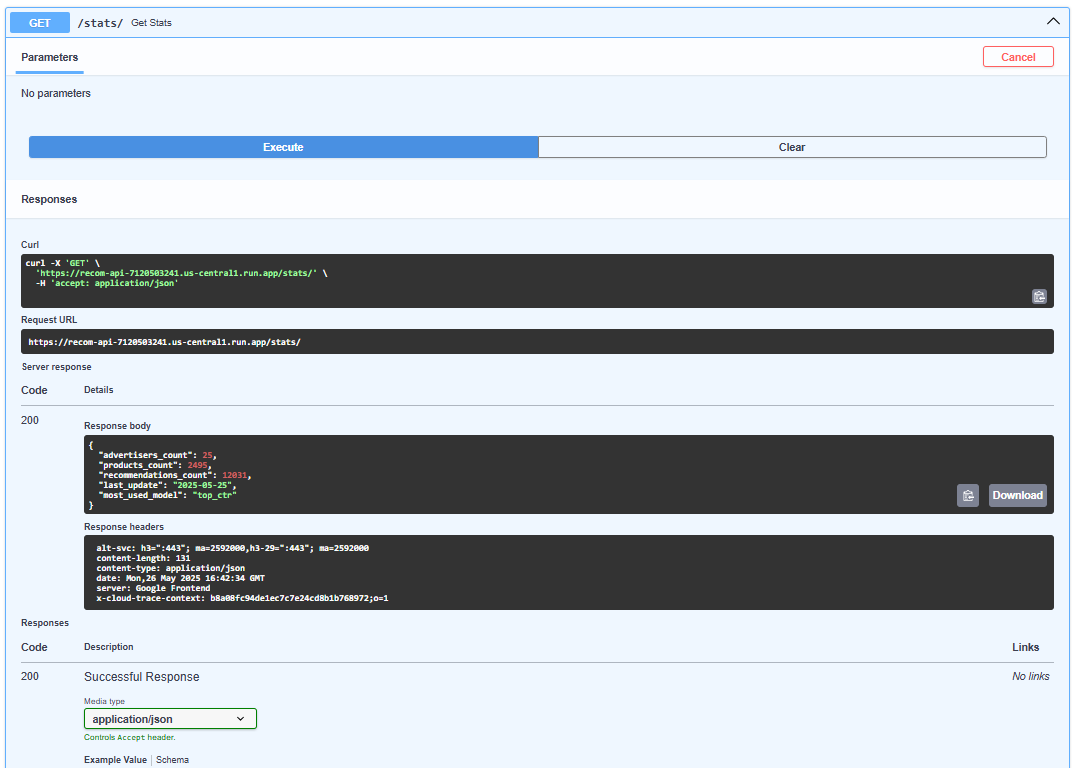
## Verificación de la API (Swagger)



## Respuestas de la API







## Docker

