

Prueba de Montaje de API con FastAPI

Para demostrar nuestra capacidad de crear una API y montarla en una instancia EC2, usamos como datos ejemplo los viajes de taxis en New York City.

Los datos fueron descargados de la página de la [ciudad](#).

Descargamos los datos en parquet y los volcamos a una base de datos SQLite.

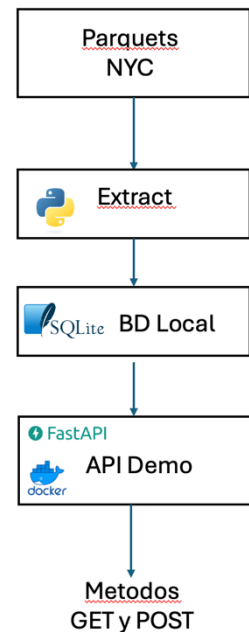
Después esa base de datos se le creó la respectiva API usando un container Dockerfile y desplegado con docker-compose.

El repositorio de la API se encuentra aquí: <https://github.com/topsy-13/LogisticaSA-CloudSolution/tree/main/03%20API%20Prueba%20Demo>

En esta URL se puede consultar la API ya desplegada y hacer peticiones GET y Post: <http://198.199.87.7:8000/docs>

A la derecha se puede ver el diagrama.

Nota: Algunas IPs difieren ya que se usó una VM en una cuenta personal después que se acabaron los créditos de AWS Academy.



Pantallazos

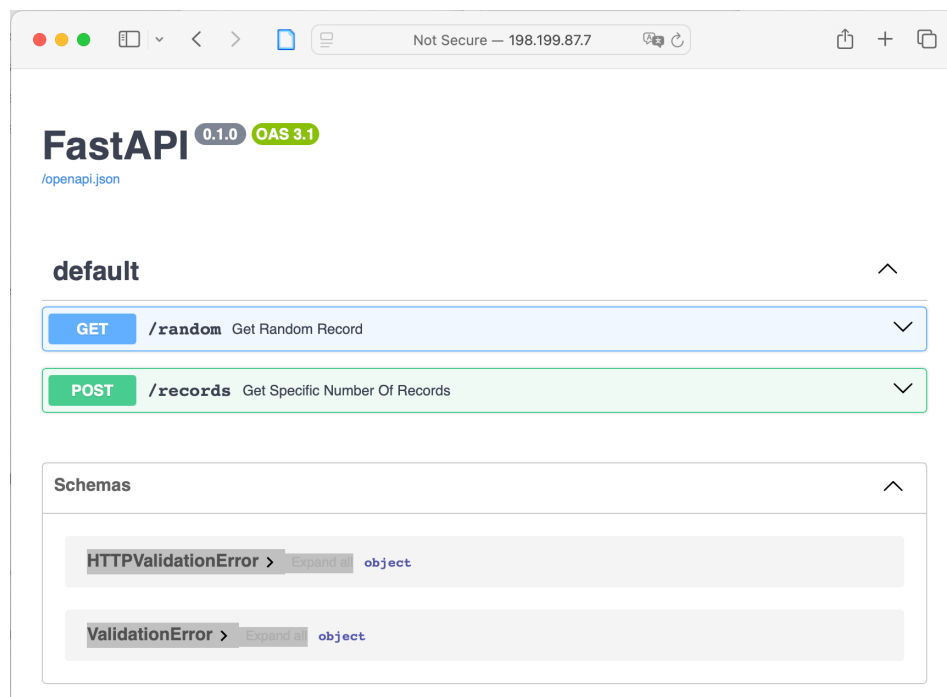


Ilustración 1 API En línea.

```

Proyecto final — ubuntu@ip-172-31-43-128: ~ — zsh — 139x41
yellow_tripdata_2022-01.parquet 26% 9690KB 166.7KB/s 02:45 E
yellow_tripdata_2022-01.parquet 30% 11MB
yellow_tripdata_2022-01.parquet 30%
yellow_tripdata_2022-01.parquet 39% 14MB 2.
yellow_tripdata_2022-01.parquet 43% 16
yellow_tripdata_2022-01.parquet 100% 36MB 131.8KB/s 04:42
nyc_taxi.db 0%
nyc_taxi.db 0%
nyc_taxi.db 5%
nyc_taxi.db 26.4
nyc_taxi.db 5% 16MB
nyc_taxi.db 100% 292MB 171.7KB/s 29:03
to_sql_lite.ipynb 100% 3306 25.6KB/s 00:00
main.cpypython-311.pyc 100% 5185 50.3KB/s 00:00
docker-compose.yml 100% 216 2.1KB/s 00:00
main.py 100% 3944 39.1KB/s 00:00
(Uni_sabana) juliantsantos@Juliants-MacBook-Pro Proyecto final %
(Uni_sabana) juliantsantos@Juliants-MacBook-Pro Proyecto final %
(Uni_sabana) juliantsantos@Juliants-MacBook-Pro Proyecto final %
(Uni_sabana) juliantsantos@Juliants-MacBook-Pro Proyecto final %
  
```

Ilustración 2. Subiendo los archivos a la instancia EC2 con SCP

Subir carpeta con el comando SCP: `scp -i etlkey.pem -r nyc-taxi-api ubuntu@172.31.43.128:/home/ubuntu/`

```

ubuntu@ip-172-31-43-128:~/nyc-taxi-api$ sudo docker-compose up --build
Creating network "nyc-taxi-api_default" with the default driver
Building api
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 344.7MB
Step 1/9 : FROM python:3.9-slim
3.9-slim: Pulling from library/python
302e3ee49805: Pull complete
4c0965d39195: Pull complete
fdeec85abba: Pull complete
62a08b8dd4f5: Pull complete
Digest: sha256:49f94609e5a997dc16086a66ac9664591854031d48e375945a9dbf4d1d53abbc
Status: Downloaded newer image for python:3.9-slim
--> 9d8cb7037cd8
Step 2/9 : WORKDIR /app
--> Running in a63bfe3a6d2e
Removing intermediate container a63bfe3a6d2e
--> ba3c7f639cd0
Step 3/9 : COPY requirements.txt requirements.txt
--> de105099007f
Step 4/9 : COPY main.py main.py
--> 00b8d169297d
Step 5/9 : COPY data_wrangling/ ./data_wrangling/
  
```

Ilustración 3. docker-compose up --build de la API