



# Estimador de ventas OPAIN

Manual de instalación y uso



**Presentado por:** Grupo 1

**Fecha:** diciembre 2, 2023

**Versión:** 1.0

# **Tabla de Contenido**

## **Introducción**

- 1.1. Propósito del manual
- 1.2. Alcance

## **Inicio**

- 2.1. Requerimientos del sistema
- 2.2. Instrucciones de instalación

## **Interfaz de usuario**

- 3.1. BONO Funcionamiento del sistema en Cloud

# 1. Introducción

## 1.1. Propósito del manual

Este manual tiene como propósito proporcionar una guía detallada y clara para utilizar eficientemente el sistema desarrollado para la predicción de ventas. Aquí encontrarás instrucciones paso a paso sobre cómo ejecutar y acceder al sistema.

## 1.2 Alcance

El manual cubre la descarga del código fuente, su ejecución y el acceso al sistema.

# 2. Inicio

## 2.1. Requerimientos del sistema

Antes de proceder con la instalación, asegúrate de que tu sistema cumple con los siguientes requisitos:

- Sistema Operativo: Linux
- RAM: 16 GB
- Espacio en disco: 20 GB
- Python 3.10
- Angular 16.2.8
- Node 18.10.0
- Docker (opcional)
- Docker compose (opcional)

## 2.2. Instrucciones de instalación

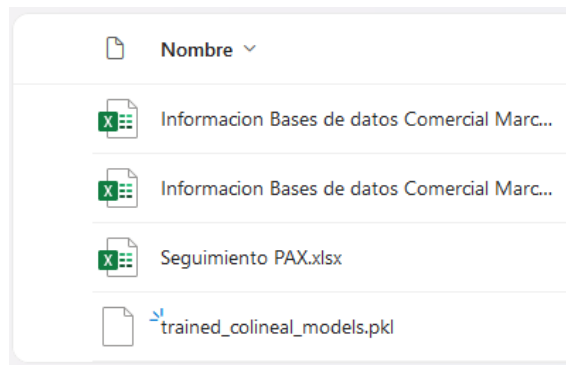
Siga las siguientes instrucciones

1. Clonar el repositorio

[https://github.com/GrupoUnoOpain/PROYECTO\\_TERCERA\\_ENTREGA](https://github.com/GrupoUnoOpain/PROYECTO_TERCERA_ENTREGA)

Por temas de privacidad de datos y debido a que el repositorio es público, no incluimos los datos que nos entregó el cliente, para que usted pueda descargarlos creamos una carpeta a la cual su usuario ya tiene acceso. Debido al peso del archivo "**trained\_colineal\_models.pkl**", este no fue cargado al repositorio, por lo cual el archivo también se puede encontrar en el directorio compartido, en el siguiente enlace:

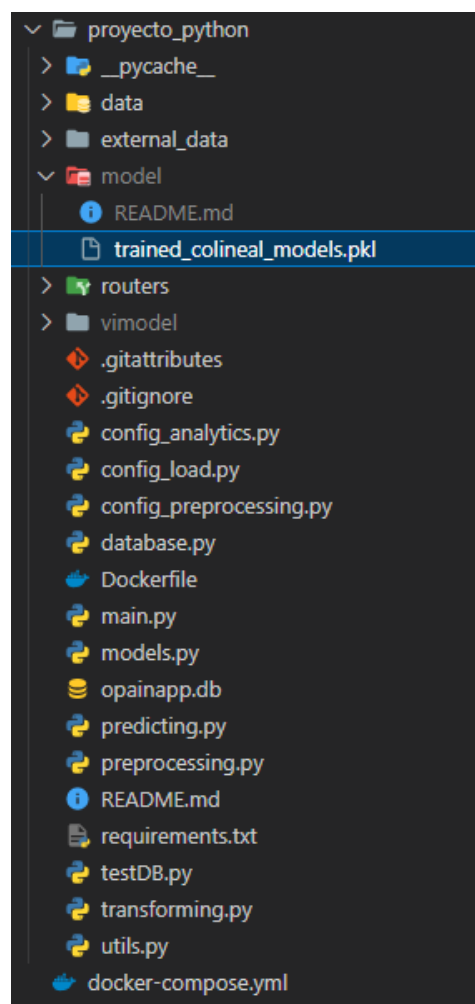
[https://uniandes-my.sharepoint.com/personal/na\\_castillo1\\_uniandes\\_edu\\_co/\\_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fna%5Fcastillo1%5Funiandes%5Fedu%5Fco%2FDocuments%2FDATOS%5FOPAIN&view=0](https://uniandes-my.sharepoint.com/personal/na_castillo1_uniandes_edu_co/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fna%5Fcastillo1%5Funiandes%5Fedu%5Fco%2FDocuments%2FDATOS%5FOPAIN&view=0)



**Figura 1. Archivos compartidos al docente**

## Backend

Después de descargar el archivo "trained\_colinear\_models.pkl", este debe ser guardado en la carpeta model del proyecto



**Figura 2. Proyecto Backend**

## Despliegue manual del backend

1. Crear el entorno virtual en la carpeta raíz del proyecto con el siguiente comando:

***Python3 -m venv vimodel***

2. Iniciar el entorno virtual con el siguiente comando:

***source vimodel/bin/activate***

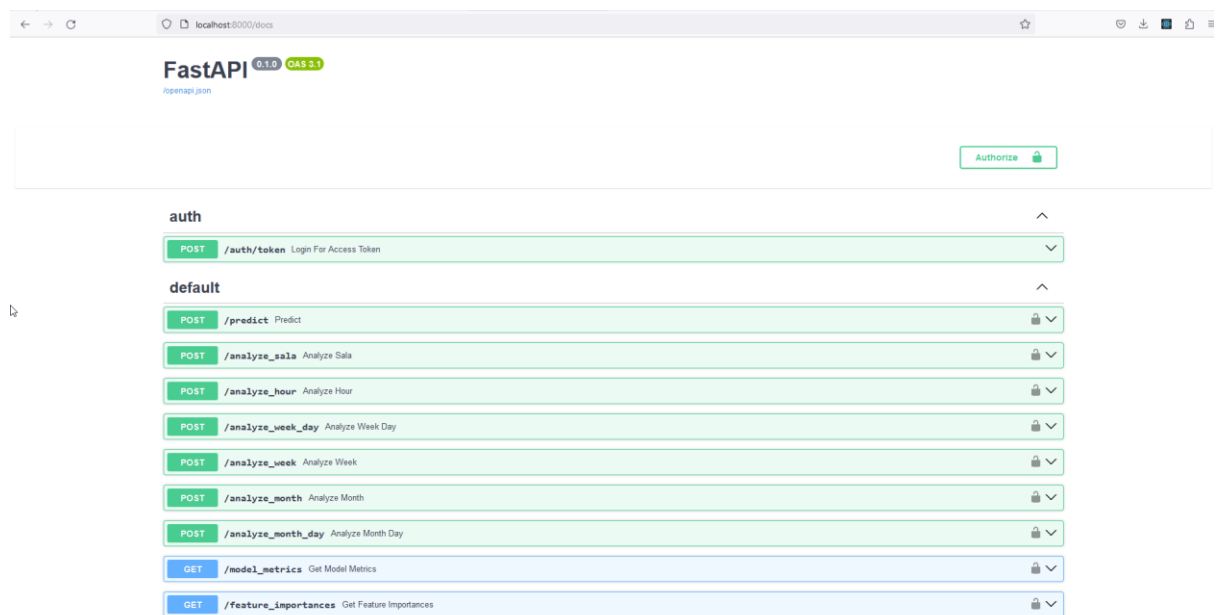
3. Instalar las dependencias haciendo uso del archivo requirements.txt que se encuentra en la raíz del proyecto, utilizando el siguiente comando:

***pip3 install -r requirements.txt***

4. Correr el programa utilizando el siguiente comando:

***uvicorn main:app***

Tan pronto el Sistema este desplegado se puede validar su funcionamiento ingresando a la siguiente url <http://localhost:8000/docs>



**Figura 3. Documentación del API REST**

Con esto finaliza el despliegue del sistema de forma manual, adicionalmente se puede desplegar el sistema haciendo uso de Docker, como se explicará a continuación.

## Despliegue del backend utilizando Docker

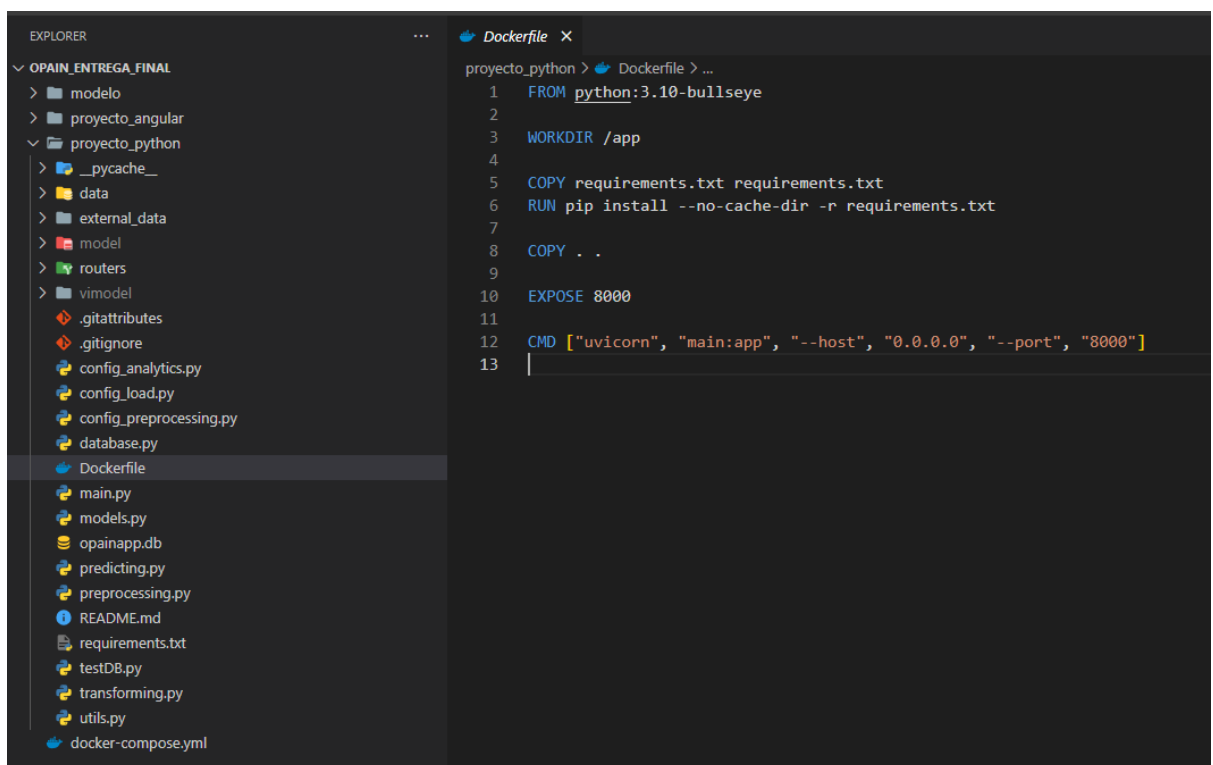
En el proyecto se agregó el archivo Dockerfile con el cual se puede crear la imagen del proyecto backend con el comando:

***docker build -t opain\_backend .***

Después se puede crear el contenedor con el siguiente comando:

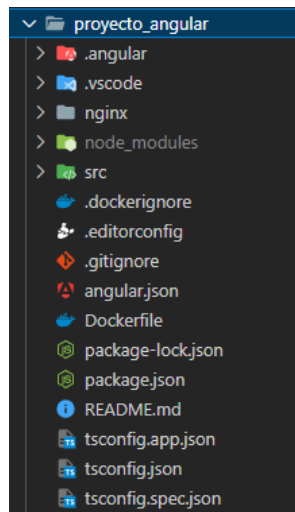
***docker run -it -p 8000:8000 opain\_backend***

Con esto se despliega la aplicación en el puerto 8000 que es el puerto seleccionado y configurado en el frontend.



**Figura 4. Dockerfile Backend**

## Frontend



**Figura 5. Proyecto Backend**

Después de descargar el código fuente del frontend, se debe acceder a la carpeta raíz y ejecutar el comando:

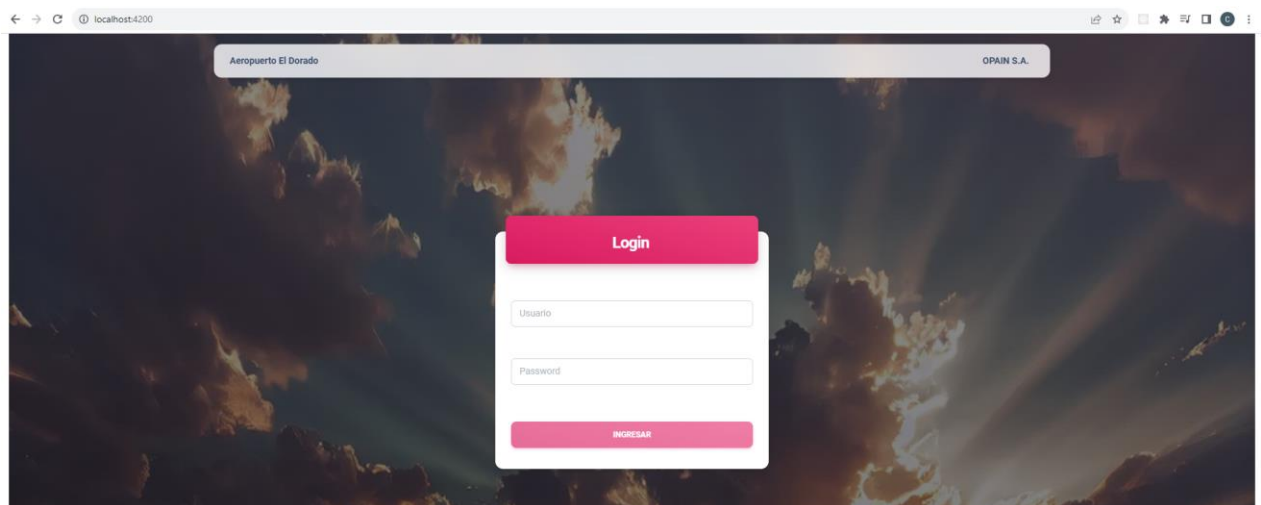
***npm install***

### Despliegue manual del frontend

Para ejecutar la aplicación se debe correr el comando:

***ng serve --open***

Tan pronto el Sistema este desplegado se puede validar su funcionamiento ingresando a la siguiente url <http://localhost:4200>



**Figura 6. Interfaz Aplicación Web**

## Despliegue del frontend utilizando Docker

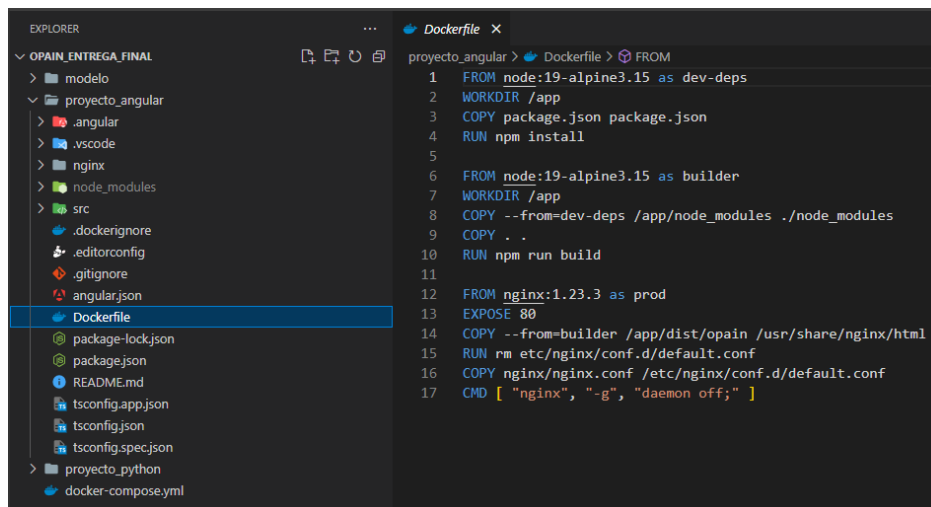
En el proyecto se agregó el archivo Dockerfile con el cual se puede crear la imagen del proyecto con el comando:

***docker build -t opain\_frontend .***

Después se puede crear el contenedor con el siguiente comando:

***docker container run -it -p 4200:4200 opain\_frontend***

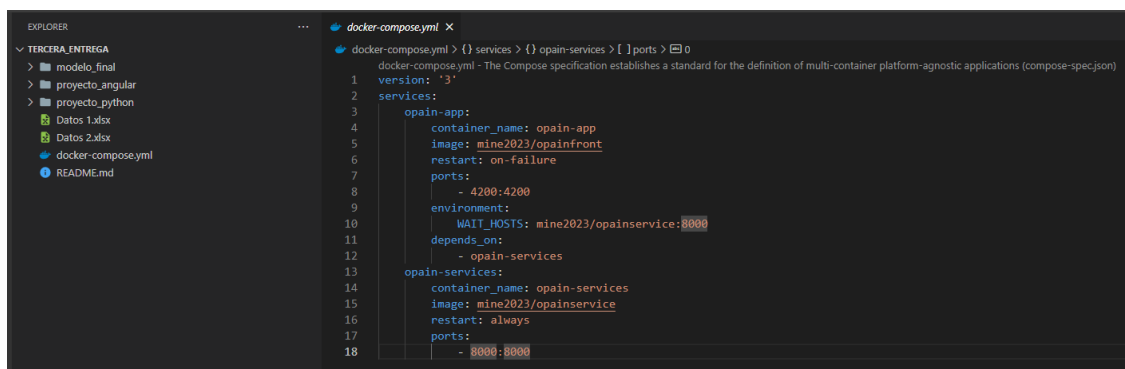
Con esto se despliega la aplicación en el puerto 4200.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'modelo', 'proyecto\_angular', and 'proyecto\_python'. The 'Dockerfile' file is selected. The code editor shows the content of the Dockerfile, which is a multi-stage build. It starts with a 'FROM' statement for 'node:19-alpine3.15 as dev-deps', followed by 'WORKDIR /app', 'COPY package.json package.json', and 'RUN npm install'. Then it switches to a second stage 'FROM node:19-alpine3.15 as builder', followed by 'WORKDIR /app', 'COPY --from=dev-deps /app/node\_modules ./node\_modules', 'COPY . .', and 'RUN npm run build'. Finally, it switches to a third stage 'FROM nginx:1.23.3 as prod', followed by 'EXPOSE 80', 'COPY --from=builder /app/dist/opain /usr/share/nginx/html', 'RUN rm etc/nginx/conf.d/default.conf', 'COPY nginx/nginx.conf /etc/nginx/conf.d/default.conf', and 'CMD [ "nginx", "-g", "daemon off;" ]'.

Figura 7. Dockerfile Backend

## DESPLIEGUE CON DOCKER COMPOSE



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'modelo\_final', 'proyecto\_angular', and 'proyecto\_python'. The 'docker-compose.yml' file is selected. The code editor shows the content of the docker-compose.yml file, which defines two services: 'opain-app' and 'opain-services'. The 'opain-app' service has a container name of 'opain-app', image of 'mine2023/opainfront', restart policy of 'on-failure', and ports '4200:4200'. The 'opain-services' service has a container name of 'opain-services', image of 'mine2023/opainservice', restart policy of 'always', and ports '8000:8000'. The 'opain-services' service depends on the 'opain-app' service.

Figura 8. Archivo docker-compose.yml

Para desplegar tanto el backend como el frontend, en la carpeta donde se haya descargado el archivo "**docker-compose.yml**", se debe ejecutar el siguiente comando que desplegara el sistema completo:



***docker-compose up***

### **3. Interfaz de usuario**

#### **3.1. BONO Funcionamiento del sistema en Cloud**

Se adjunta el video de YouTube el cual muestra el funcionamiento el sistema desplegado en el proveedor Cloud AWS.



**Video 1. Funcionamiento del sistema en Cloud**