



MIAD

Maestría
en Inteligencia
Analítica de Datos



Taller APIs

Instrucciones:

En este taller exploraremos el servicio de predicciones de modelos de analítica a través de una API.

La entrega de este taller consiste en un reporte en formato de documento de texto, donde pueda incorporar fácilmente capturas de pantalla, textos y elementos similares. Puede utilizar formatos como Word, LibreOffice, Markdown, u otros.

Parte 1: exploración de los archivos de una API

1. Junto a este enunciado encontrará el archivo *bankchurn-api.zip*.
2. Descomprima este archivo localmente y explore la estructura de carpetas y archivos.
3. Explore los archivos de requirements que encontrará en la carpeta raíz, que contienen las dependencias del paquete (*requirements.txt*), las de prueba (*test_requirements.txt*) y las de tipos (*typing_requirements.txt*).
4. Abra el archivo *tox.ini* y familiarícese con su estructura y contenido. En este archivo se definen dos ambientes clave que usaremos hacia adelante: *testapp* y *run*. En cada uno de estos ambientes se define un directorio para el ambiente, un conjunto de dependencias y los comandos a ejecutar. **Incluya en su reporte una descripción de qué dependencias requiere cada uno de estos ambientes y qué comandos ejecuta.**
5. En la carpeta *app/schemas* encontrará dos esquemas para la API: *health* y *predict*. **Incluya en su reporte una descripción de los dos esquemas definidos allí. Explique qué contenidos incluyen estos esquemas.**
6. En la carpeta *app* encontrará el archivo principal de la api, *main.py*, donde se ejecuta el servidor *uvicorn*. **Incluya en su reporte una descripción de los 4 argumentos de la función *run* empleados allí.**
7. Explore el resto del script *main.py*. Note que se define la ruta raíz y la respuesta a una solicitud a esta ruta en formato HTML.



8. En el archivo `config.py` podrá encontrar algunos pasos de configuración. En la clase `Settings` se define la ruta base, el nombre del proyecto y se habilita el acceso CORS desde varios orígenes. Aquí también se define la configuración de los loggers, que registran la ejecución de la API.
9. Finalmente, el corazón de la ejecución de las rutas de la API se encuentra en `api.py`. Allí encontrará las dos rutas definidas: `health` y `predict`. **Incluya en su reporte una descripción de qué se ejecuta cuando se hace un llamado a cada una de estas rutas.**

Parte 2: probando y ejecutando la API

1. Lance una instancia en AWS EC2. Se recomienda una máquina `t2.small`, con sistema operativo Ubuntu y 20GB de disco. Incluya un pantallazo de la consola de AWS EC2 con la máquina en ejecución en su reporte. Su usuario de AWS debe estar visible en el pantallazo.
2. Suba el archivo `bankchurn-api.zip` a la máquina con un comando como

```
scp -i llave.pem bankchurn-api.zip ubuntu@IP:/home/ubuntu
```

3. Conéctese a la máquina:
 - a) Abra una terminal: En windows, escriba `cmd` y `Enter`. En macOS, abra la aplicación llamada *Terminal*.
 - b) En la terminal emita el comando

```
ssh -i /path/to/llave.pem ubuntu@IP
```

donde `/path/to/` se refiere a la ubicación del archivo `llave.pem` que descargó, e `IP` es la dirección IP de la instancia EC2 que lanzó. Si prefiere, en la terminal puede navegar a la ubicación del archivo `llave.pem` y emitir el comando

```
ssh -i llave.pem ubuntu@IP
```

Note que usamos en este caso `ubuntu`, pues éste es el usuario creado por defecto como administrador con sistema operativo Ubuntu server. Incluya en su reporte un *screenshot* de la conexión a la máquina virtual.

4. Actualice las ubicaciones de los paquetes con el siguiente comando:

```
sudo apt update
```

5. Instale pip para python3

```
sudo apt install python3-pip
```

6. Instale unzip para manipular los archivos `.zip`

```
sudo apt install zip unzip
```



7. Instale venv para crear ambientes virtuales con el siguiente comando:

```
sudo apt install python3.12-venv
```

8. Cree un ambiente virtual con el nombre env-api

```
python3 -m venv /home/ubuntu/env-api
```

9. Active el ambiente con el comando:

```
source env-api/bin/activate
```

10. Descomprima el archivo bankchurn-api.zip

```
unzip bankchurn-api.zip
```

Verifique que haya quedado la carpeta bankchurn-api creada

11. Instale tox, una librería de automatización y pruebas

```
pip install tox
sudo apt-get install tox
```

Incluya la dirección /home/ubuntu/.local/bin en el PATH para facilitar su ejecución

```
PATH=$PATH:/home/ubuntu/.local/bin
```

12. Ingrese a la carpeta bankchurn-api

```
cd bankchurn-api
```

y ejecute el ambiente de prueba de la api de tox

```
tox run -e test_app
```

Incluya en su reporte un *screenshot* de la salida de este comando y una breve explicación.

Verifique que se pasen todas las pruebas.

13. Ejecute ahora el ambiente de ejecución de tox

```
tox run -e run
```

Incluya en su reporte un *screenshot* de la salida de este comando y una breve explicación.

Verifique que se pasen todas las pruebas.

14. Como resultado del anterior comando, el servidor Unicorn debe estar en ejecución y su API debe estar disponible en el puerto 8001 de su máquina virtual. Abra el puerto 8001 en el grupo de seguridad de la máquina virtual, así:

- En la consola de EC2, panel izquierdo, seleccione Security Groups en redes y seguridad.
- Seleccione el grupo de seguridad correspondiente a su máquina virtual (puede encontrar el nombre en los detalles de la instancia).



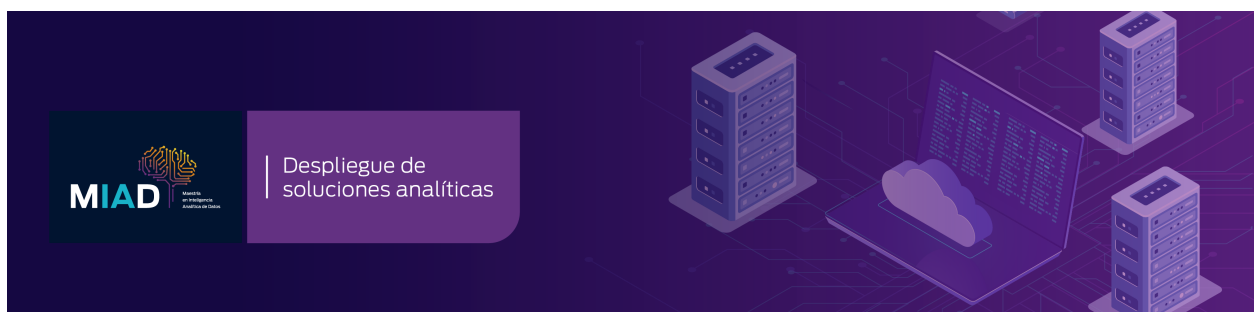
- c) Click en Editar reglas de entrada.
 - d) Click en Agregar regla. En Tipo seleccione TCP personalizado, en Intervalo de puertos marque 8001, y en Origin seleccione Anywhere IPv4. Click en Guardar regla.
15. Acceda a la API en `http://IP:8001`, donde IP es la IP pública de su máquina virtual.
16. Interactúe con la API y sus dos rutas, la de health y la de predicción. **Incluya en su reporte un *screenshot* donde se evidencie la ejecución de la API en el navegador y la respuesta satisfactoria de una solicitud a la ruta de predicción.**

Parte 3: modificando la API

1. Para profundizar un poco más en el empaquetamiento de modelos y API, continuemos con el ejercicio de modificación del paquete de la actividad anterior. Allí se eliminó una variable (features) del modelo y se creó una nueva versión del paquete. El archivo correspondiente deberá tener el nombre `"model_abandono-0.0.2-py3-none-any.whl"`
2. Modifique la API para incorporar este cambio. Primero, localmente, reemplace el archivo `"model_abandono-0.0.2-py3-none-any.whl"` que se encuentra en la carpeta `model-package` por su correspondiente versión 0.0.2.
3. Localmente, modifique el archivo `requirements.txt` para que la versión del paquete corresponda con la 0.0.2.
4. Localmente, comprima la carpeta con las modificaciones. Adicionalmente, renombre la carpeta como `"bankchurn-api2.zip"`.
5. Utilizando un repositorio en Git, realice el commit correspondiente para subir la nueva carpeta comprimida. Para ello deberá crear una carpeta local nueva que contenga a `"bankchurn-api2.zip"`, utilizar git para agregar el archivo, realizar el respectivo commit, enlazar este repositorio con un nuevo repositorio en línea en GitHub, enlazarlo con su máquina local y realizar el push para que quede en la rama master remota. Las líneas de código se muestran a continuación asumiendo que abre la terminal local desde la carpeta local donde tiene el archivo comprimido y el URL del repositorio en GitHub es URL:

```
git init
git add bankchurn-api2.zip
git commit -m "API modificada"
git remote add origin URL
git push origin master
```

Incluya en su reporte un pantallazo del archivo `bankchurn-api2.zip` en GitHub.



6. Desde su máquina virtual, asegúrese de que Git esté instalado utilizando el siguiente comando:

```
sudo apt install git
```

7. Clone el repositorio con la nueva versión del paquete. Para ello utilice el siguiente comando, donde URL corresponde al enlace HTTPS del repositorio en GitHub:

```
git clone URL
```

8. Ingrese a la carpeta clonada utilizando el siguiente comando asumiendo que su repositorio en GitHub tiene el nombre my-repo:

```
cd my-repo
```

9. Descomprima el archivo utilizando el siguiente comando:

```
unzip bankchurn-api2.zip
```

10. Repita los pasos de prueba para verificar que pasen adecuadamente.
11. Repita los pasos de ejecución de la API y pruebe su funcionamiento en el navegador.
12. Con la API en ejecución, edite el archivo de inputs para la predicción eliminando una de las variables que **SÍ** eliminó en la versión 0.0.2 del paquete. **Incluya en su reporte un pantallazo del resultado de ejecutar la predicción una vez se hace este cambio.**
13. Con la API en ejecución, edite el archivo de inputs para la predicción eliminando una de las variables que **NO** eliminó en la versión 0.0.2 del paquete. **Incluya en su reporte un pantallazo del resultado de ejecutar la predicción una vez se hace este cambio.**