# Documentación sobre el Servidor.

## Conexión con el servidor.

El servidor puede estar disponible para la conexión de varias formas:

* Únicamente en local.
* Con acceso para la red Wifi.
* Con acceso para todo internet.

### En local

Corremos el servidor con el comando:

uvicorn server:app

La url de acceso será http://127.0.0.1:8000/XXXX

### Con acceso para la red wifi:

Corremos el servidor con el comando:

uvicorn server:app --host 0.0.0.0 --port 8000

Entonces, si nuestra ip es 10.194.10.252, la url de acceso será: http://10.194.10.252.local:8000/XXXX

Se puede hacer más fácilmente usando Bonjour services, que permite usar el nombre del host como ip. Tras descargar Bonjour para windows, escribimos en cmd el comando.   
hostname

Entonces recibiremos el nombre, en mi caso, ANT-VICTORF.

La url de acceso será:

<http://ANT-VICTORF.local:8000/XXXX>

EN ANDROID HAY QUE DESCARGAR ALGUNA APP QUE PERMITA USAR EL NOMBRE COMO IP, COMO SERVICE BROWSER

### Con acceso para todo internet:

Preparar NGROK (Instalar, añadir al path, etc).

Corremos el servidor con alguno de los comandos vistos anteriormente (local o red wifi)

En otro terminal corremos el comando:  
ngrok http 8000

Nos aparecerá una url, algo así como <https://preexistent-lou-xylographic.ngrok-free.dev>.

La url de nuestro servidor será https://preexistent-lou-xylographic.ngrok-free.dev/XXXX

## Carpeta de modelos

En el servidor existe la carpeta “models”, que contiene las carpetas de aquellos modelos disponibles. Cada modelo tiene que tener una carpeta que contenga los archivos:

* Classifier\_(model\_name).keras
* Encoder\_(model\_name).pkl
* Preprocesa\_(model\_name).py
* Scaler\_(model\_name).pkl

Donde model\_name es el nombre del modelo y de la carpeta que contiene los archivos.

## Ejecución del servidor en la nube

### Render.com

Una opción gratuita para ejecutar el servidor la ofrece la web Render. El código del servidor debe estar en un repositorio github. Guía explicativa: <https://render.com/docs/deploy-fastapi>.

Problema: En la versión gratuita, a los 15 minutos de no recibir ninguna request, el servicio web se cae automáticamente y tardará unos minutos en reinicializarse cuando mandemos la siguiente request. Para solucionar eso mandamos un mensaje de monitorización cada 12 minutos con la web uptime Robot. Parece que funciona bien.

Por lo demás, muy sencillo de utilizar. La url web será, por ejemplo: "https://snifflinkserver.onrender.com"