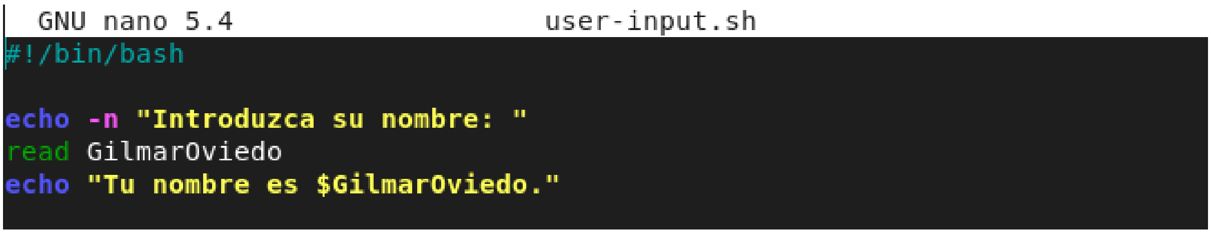
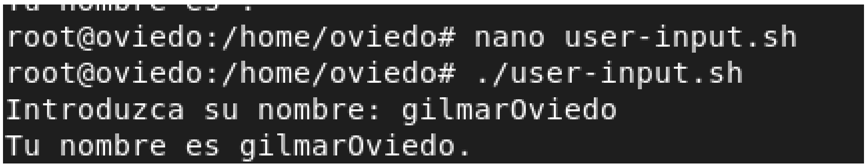
1. Crear un Bash Script simple

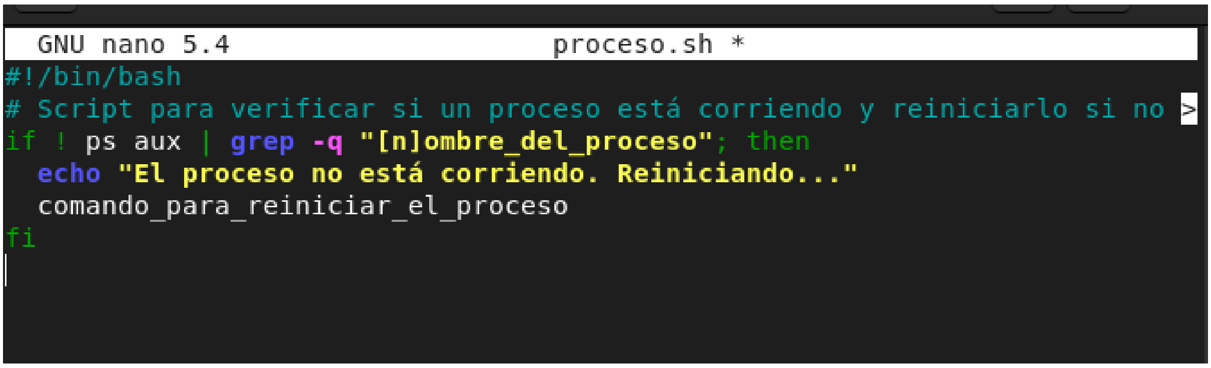


```
  GNU nano 5.4                          user-input.sh
#!/bin/bash

echo -n "Introduzca su nombre: "
read GilmarOviedo
echo "Tu nombre es $GilmarOviedo."
```

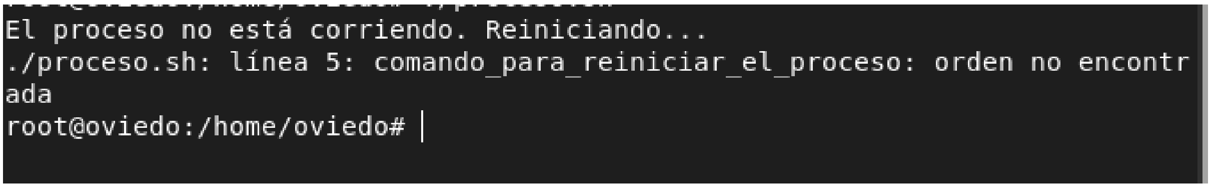


```
root@oviedo:/home/oviedo# nano user-input.sh
root@oviedo:/home/oviedo# ./user-input.sh
Introduzca su nombre: gilmarOviedo
Tu nombre es gilmarOviedo.
```

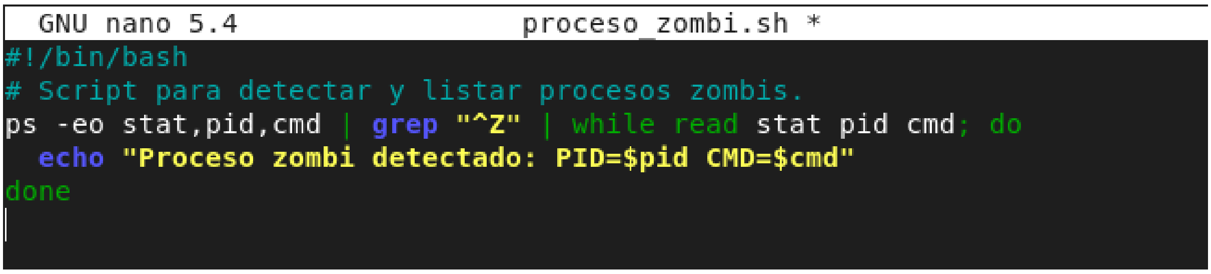
proceso



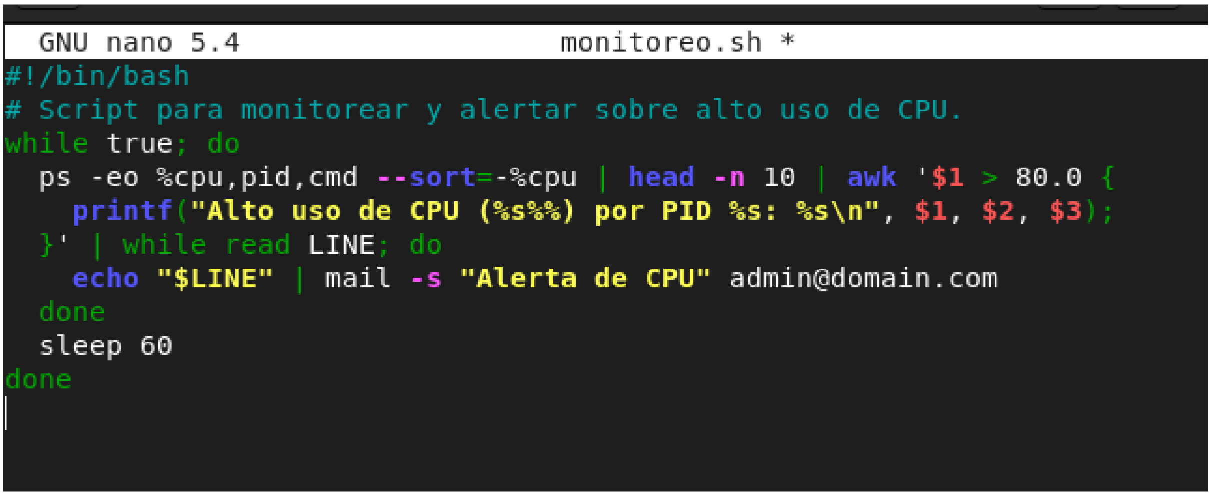
```
  GNU nano 5.4                             proceso.sh *
#!/bin/bash
# Script para verificar si un proceso está corriendo y reiniciarlo si no >
if ! ps aux | grep -q "[n]ombre_del_proceso"; then
  echo "El proceso no está corriendo. Reiniciando..."
  comando_para_reiniciar_el_proceso
fi
```



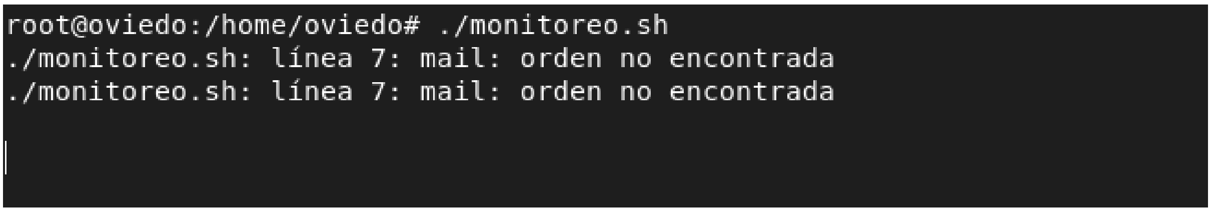
```
El proceso no está corriendo. Reiniciando...
./proceso.sh: línea 5: comando_para_reiniciar_el_proceso: orden no encontr
ada
root@oviedo:/home/oviedo# 
```



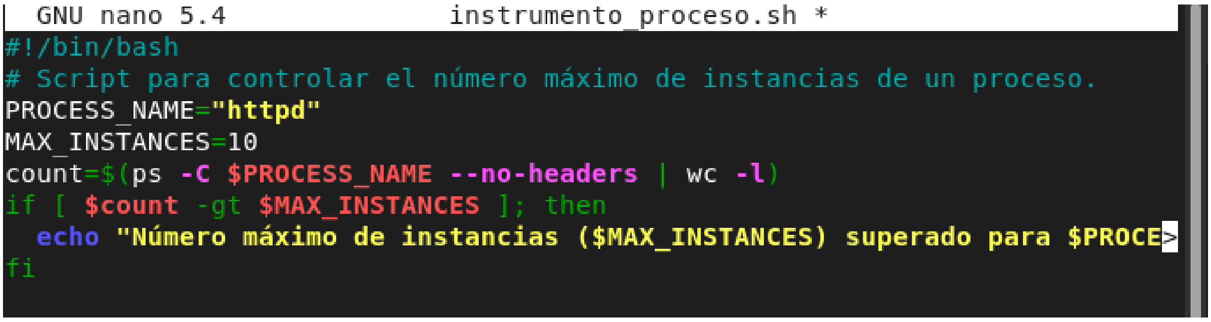
```
  GNU nano 5.4                          proceso_zombi.sh *
#!/bin/bash
# Script para detectar y listar procesos zombis.
ps -eo stat,pid,cmd | grep "^Z" | while read stat pid cmd; do
  echo "Proceso zombi detectado: PID=$pid CMD=$cmd"
done
```



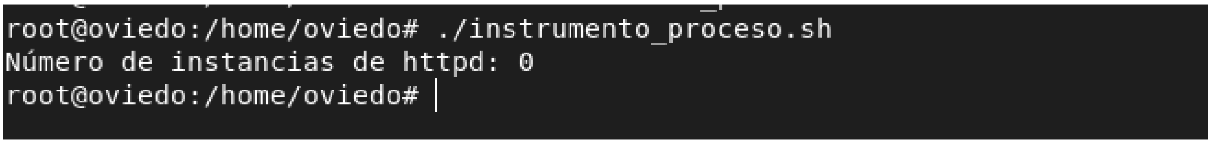
```
  GNU nano 5.4                        monitoreo.sh *
#!/bin/bash
# Script para monitorear y alertar sobre alto uso de CPU.
while true; do
  ps -eo %cpu,pid,cmd --sort=-%cpu | head -n 10 | awk '$1 > 80.0 {
    printf("Alto uso de CPU (%s%%) por PID %s: %s\n", $1, $2, $3);
  }' | while read LINE; do
    echo "$LINE" | mail -s "Alerta de CPU" admin@domain.com
  done
  sleep 60
done
```



```
root@oviedo:/home/oviedo# ./monitoreo.sh
./monitoreo.sh: línea 7: mail: orden no encontrada
./monitoreo.sh: línea 7: mail: orden no encontrada
```

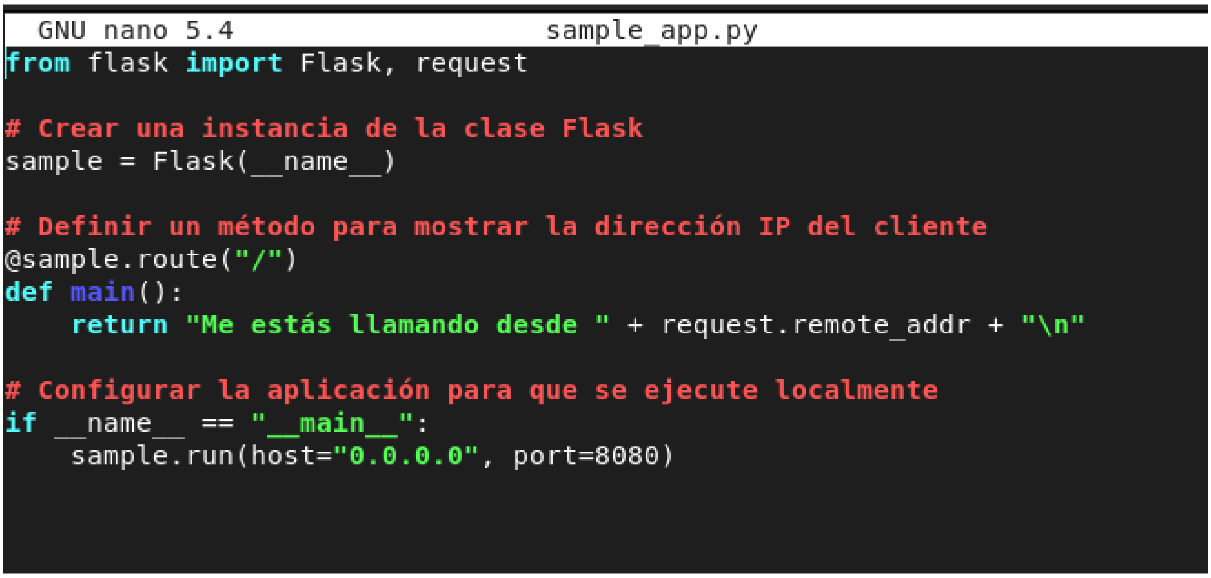


```
  GNU nano 5.4                  instrumento_proceso.sh *
#!/bin/bash
# Script para controlar el número máximo de instancias de un proceso.
PROCESS_NAME="httpd"
MAX_INSTANCES=10
count=$(ps -C $PROCESS_NAME --no-headers | wc -l)
if [ $count -gt $MAX_INSTANCES ]; then
  echo "Número máximo de instancias ($MAX_INSTANCES) superado para $PROCE>
fi
```

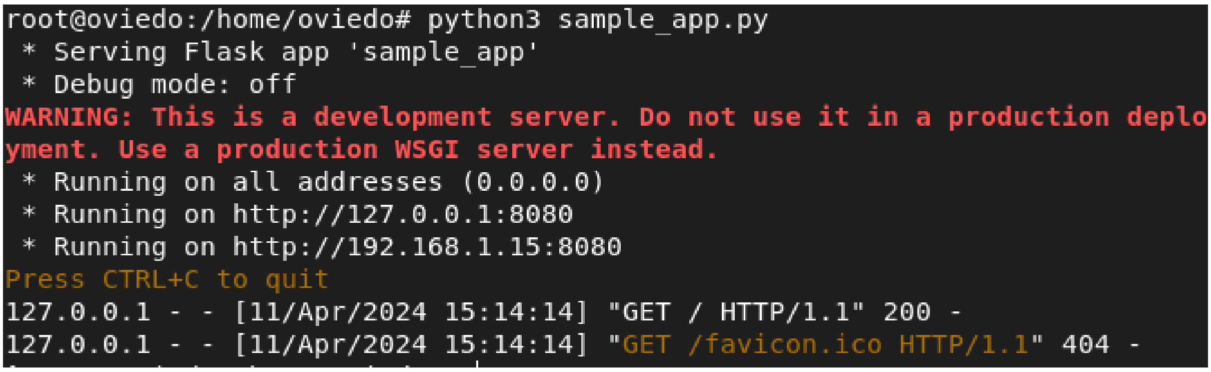


```
root@oviedo:/home/oviedo# ./instrumento_proceso.sh
Número de instancias de httpd: 0
root@oviedo:/home/oviedo#
```

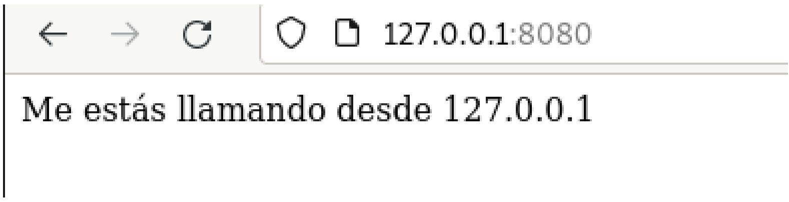
2.-Crear una aplicación web simple



```
  GNU nano 5.4                              sample_app.py
from flask import Flask, request

# Crear una instancia de la clase Flask
sample = Flask(__name__)

# Definir un método para mostrar la dirección IP del cliente
@sample.route("/")
def main():
    return "Me estás llamando desde " + request.remote_addr + "\n"

# Configurar la aplicación para que se ejecute localmente
if __name__ == "__main__":
    sample.run(host="0.0.0.0", port=8080)
```

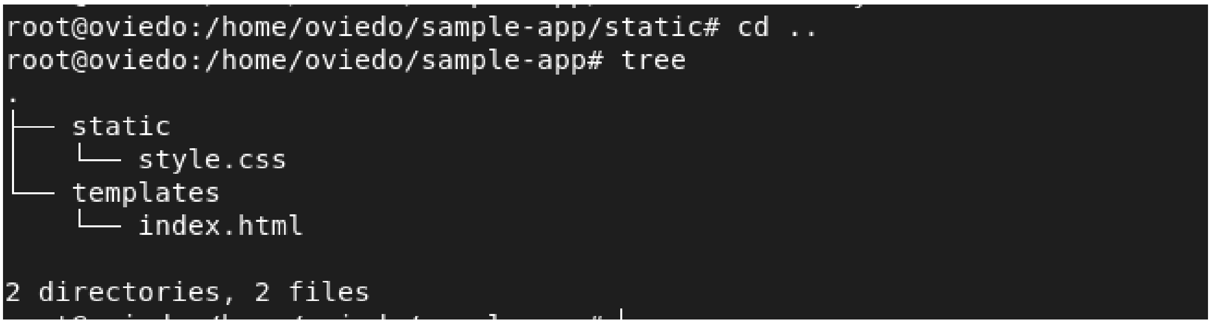


```
root@oviedo:/home/oviedo# python3 sample_app.py
 * Serving Flask app 'sample_app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deplo
yment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://192.168.1.15:8080
Press CTRL+C to quit
127.0.0.1 - - [11/Apr/2024 15:14:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Apr/2024 15:14:14] "GET /favicon.ico HTTP/1.1" 404 -
```

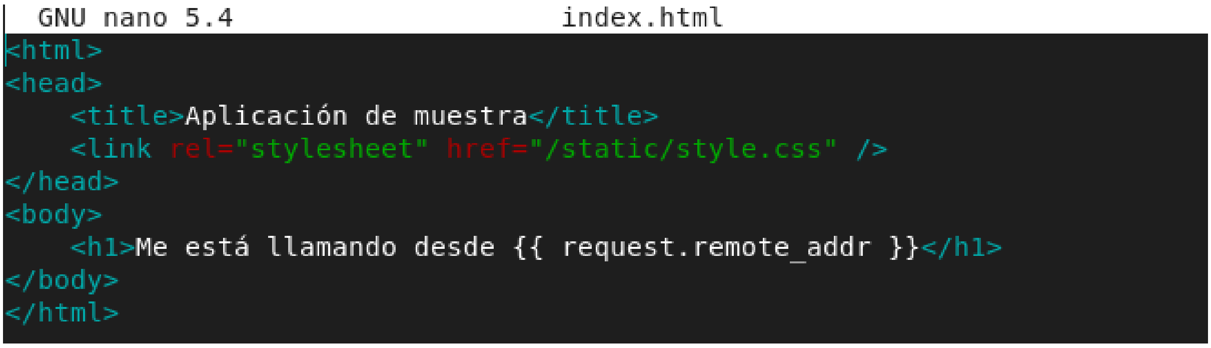


Me estás llamando desde 127.0.0.1

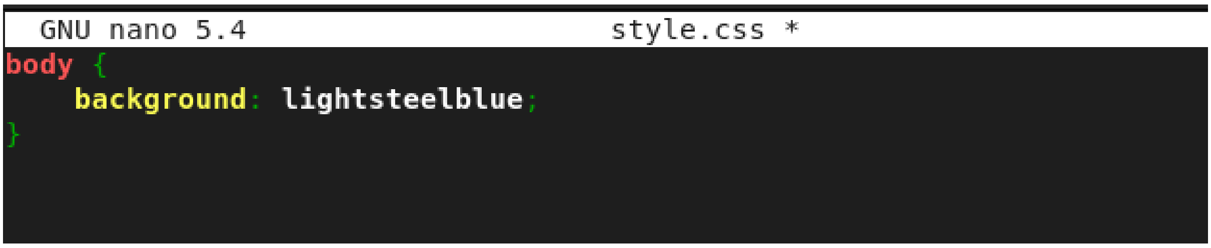
3.-Configurar la aplicación web para utilizar archivos de sitio web



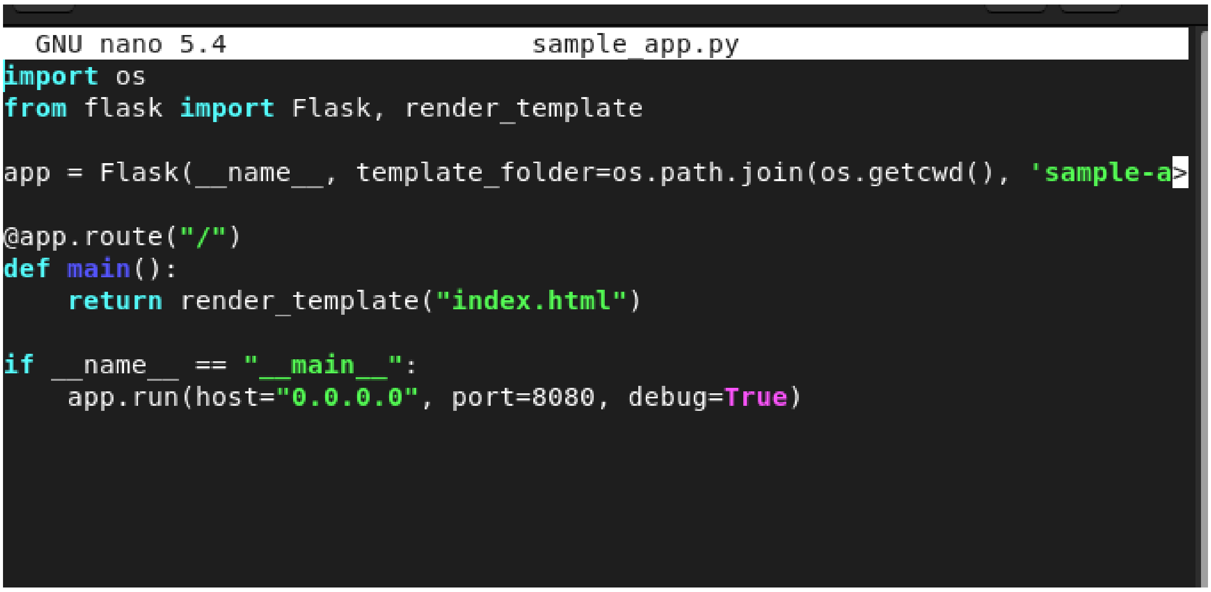
```
root@oviedo:/home/oviedo/sample-app/static# cd ..
root@oviedo:/home/oviedo/sample-app# tree
.
├── static
│   └── style.css
└── templates
    └── index.html

2 directories, 2 files
```



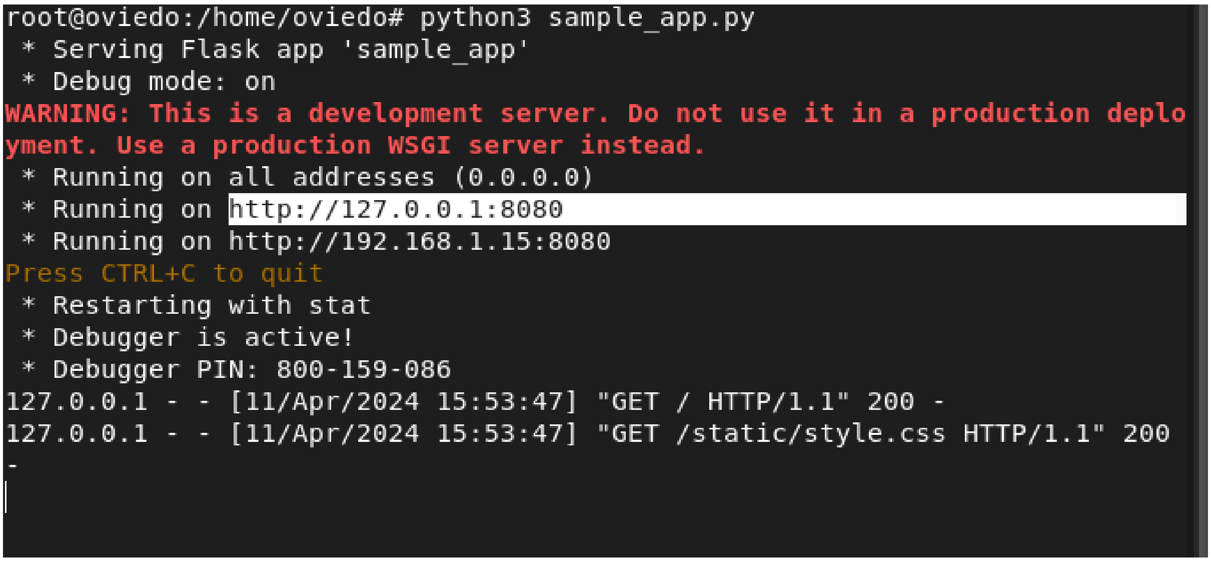
```
  GNU nano 5.4                       index.html
<html>
<head>
    <title>Aplicación de muestra</title>
    <link rel="stylesheet" href="/static/style.css" />
</head>
<body>
    <h1>Me está llamando desde {{ request.remote_addr }}</h1>
</body>
</html>
```



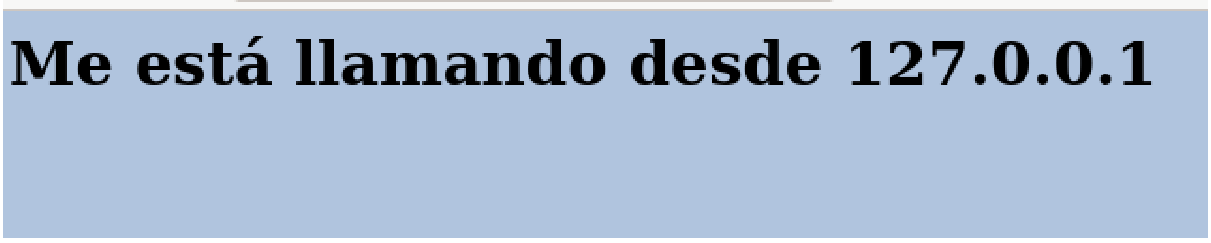
```
  GNU nano 5.4                         style.css *
body {
    background: lightsteelblue;
}
```



```
  GNU nano 5.4                       sample_app.py
import os
from flask import Flask, render_template

app = Flask(__name__, template_folder=os.path.join(os.getcwd(), 'sample-a>

@app.route("/")
def main():
    return render_template("index.html")

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8080, debug=True)
```

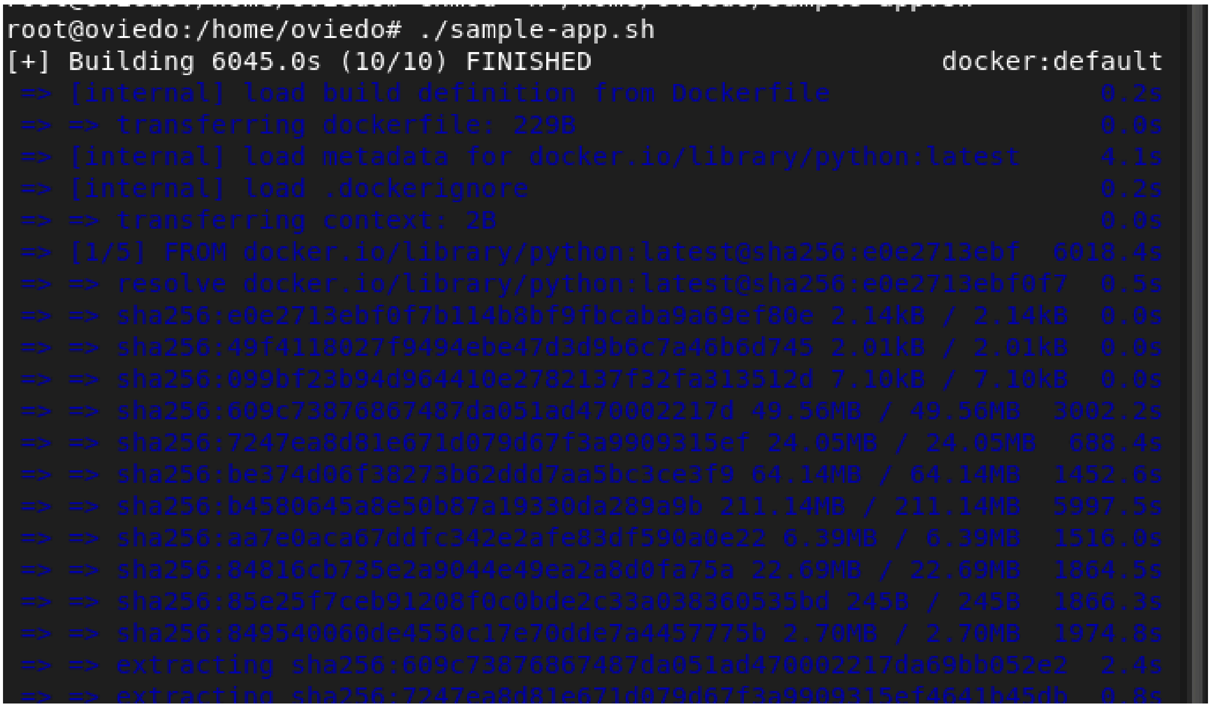


```
root@oviedo:/home/oviedo# python3 sample_app.py
 * Serving Flask app 'sample_app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deplo
yment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://192.168.1.15:8080
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 800-159-086
127.0.0.1 - - [11/Apr/2024 15:53:47] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Apr/2024 15:53:47] "GET /static/style.css HTTP/1.1" 200
-
```

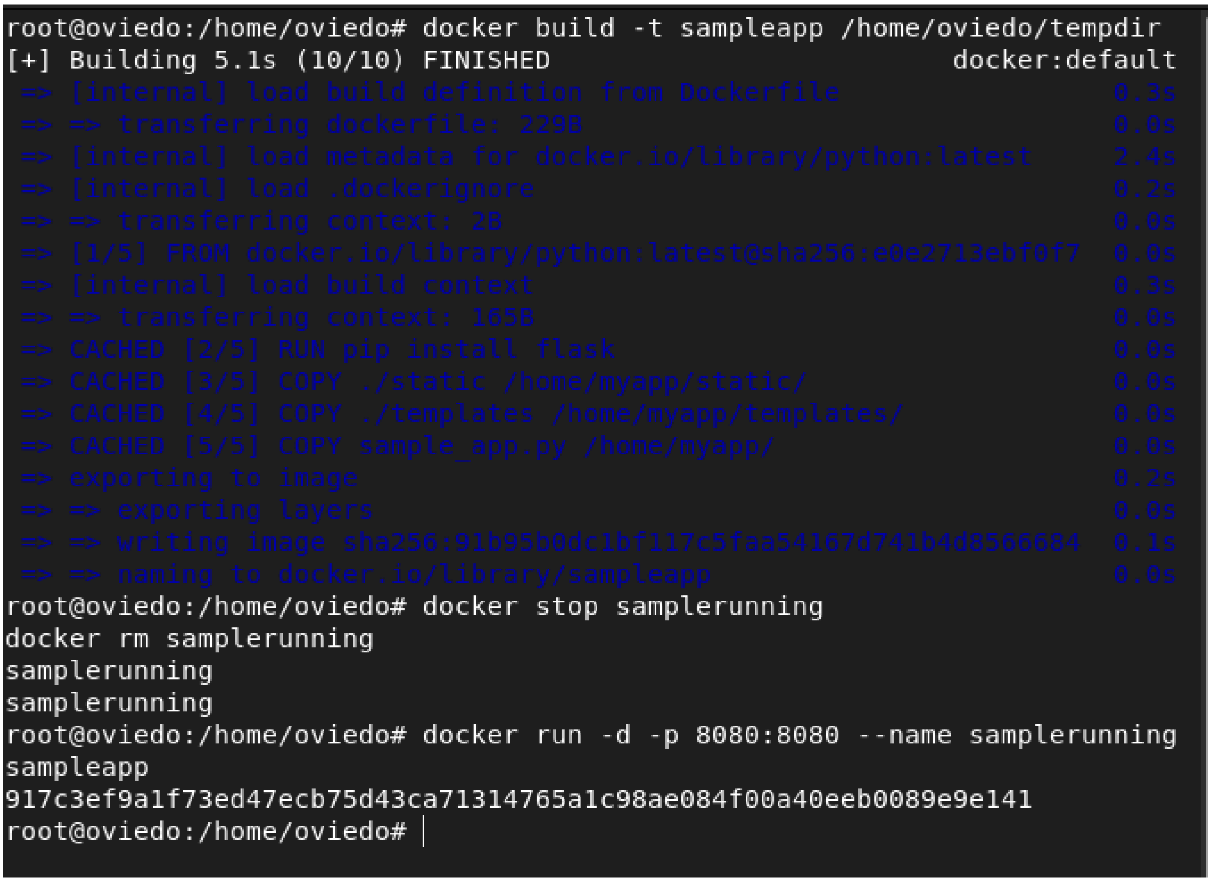


Me está llamando desde 127.0.0.1

4.-Crear un script de Bash para compilar y ejecutar un contenedor Docker



```
root@oviedo:/home/oviedo# ./sample-app.sh
[+] Building 6045.0s (10/10) FINISHED                            docker:default
 => [internal] load build definition from Dockerfile                        0.2s
 => => transferring dockerfile: 229B                                        0.0s
 => [internal] load metadata for docker.io/library/python:latest            4.1s
 => [internal] load .dockerignore                                           0.2s
 => => transferring context: 2B                                             0.0s
 => [1/5] FROM docker.io/library/python:latest@sha256:e0e2713ebf         6018.4s
 => => resolve docker.io/library/python:latest@sha256:e0e2713ebf0f7         0.5s
 => => sha256:e0e2713ebf0f7b114b8bf9fbcaba9a69ef80e 2.14kB / 2.14kB         0.0s
 => => sha256:49f4118027f9494ebe47d3d9b6c7a46b6d745 2.01kB / 2.01kB         0.0s
 => => sha256:099bf23b94d964410e2782137f32fa313512d 7.10kB / 7.10kB         0.0s
 => => sha256:609c73876867487da051ad470002217d 49.56MB / 49.56MB         3002.2s
 => => sha256:7247ea8d81e671d079d67f3a9909315ef 24.05MB / 24.05MB         688.4s
 => => sha256:be374d06f38273b62ddd7aa5bc3ce3f9 64.14MB / 64.14MB         1452.6s
 => => sha256:b4580645a8e50b87a19330da289a9b 211.14MB / 211.14MB         5997.5s
 => => sha256:aa7e0aca67ddfc342e2afe83df590a0e22 6.39MB / 6.39MB         1516.0s
 => => sha256:84816cb735e2a9044e49ea2a8d0fa75a 22.69MB / 22.69MB         1864.5s
 => => sha256:85e25f7ceb91208f0c0bde2c33a038360535bd 245B / 245B         1866.3s
 => => sha256:849540060de4550c17e70dde7a4457775b 2.70MB / 2.70MB         1974.8s
 => => extracting sha256:609c73876867487da051ad470002217da69bb052e2         2.4s
 => => extracting sha256:7247ea8d81e671d079d67f3a9909315ef4641b45db         0.8s
```



```
root@oviedo:/home/oviedo# docker build -t sampleapp /home/oviedo/tempdir
[+] Building 5.1s (10/10) FINISHED                              docker:default
 => [internal] load build definition from Dockerfile                      0.3s
 => => transferring dockerfile: 229B                                      0.0s
 => [internal] load metadata for docker.io/library/python:latest          2.4s
 => [internal] load .dockerignore                                         0.2s
 => => transferring context: 2B                                           0.0s
 => [1/5] FROM docker.io/library/python:latest@sha256:e0e2713ebf0f7       0.0s
 => [internal] load build context                                         0.3s
 => => transferring context: 165B                                         0.0s
 => CACHED [2/5] RUN pip install flask                                    0.0s
 => CACHED [3/5] COPY ./static /home/myapp/static/                        0.0s
 => CACHED [4/5] COPY ./templates /home/myapp/templates/                  0.0s
 => CACHED [5/5] COPY sample_app.py /home/myapp/                          0.0s
 => exporting to image                                                    0.2s
 => => exporting layers                                                   0.0s
 => => writing image sha256:91b95b0dc1bf117c5faa54167d741b4d8566684       0.1s
 => => naming to docker.io/library/sampleapp                              0.0s
root@oviedo:/home/oviedo# docker stop samplerunning
docker rm samplerunning
samplerunning
samplerunning
root@oviedo:/home/oviedo# docker run -d -p 8080:8080 --name samplerunning
sampleapp
917c3ef9a1f73ed47ecb75d43ca71314765a1c98ae084f00a40eeb0089e9e141
root@oviedo:/home/oviedo# 
```

5.-Crear, ejecutar y verificar el contenedor Docker