

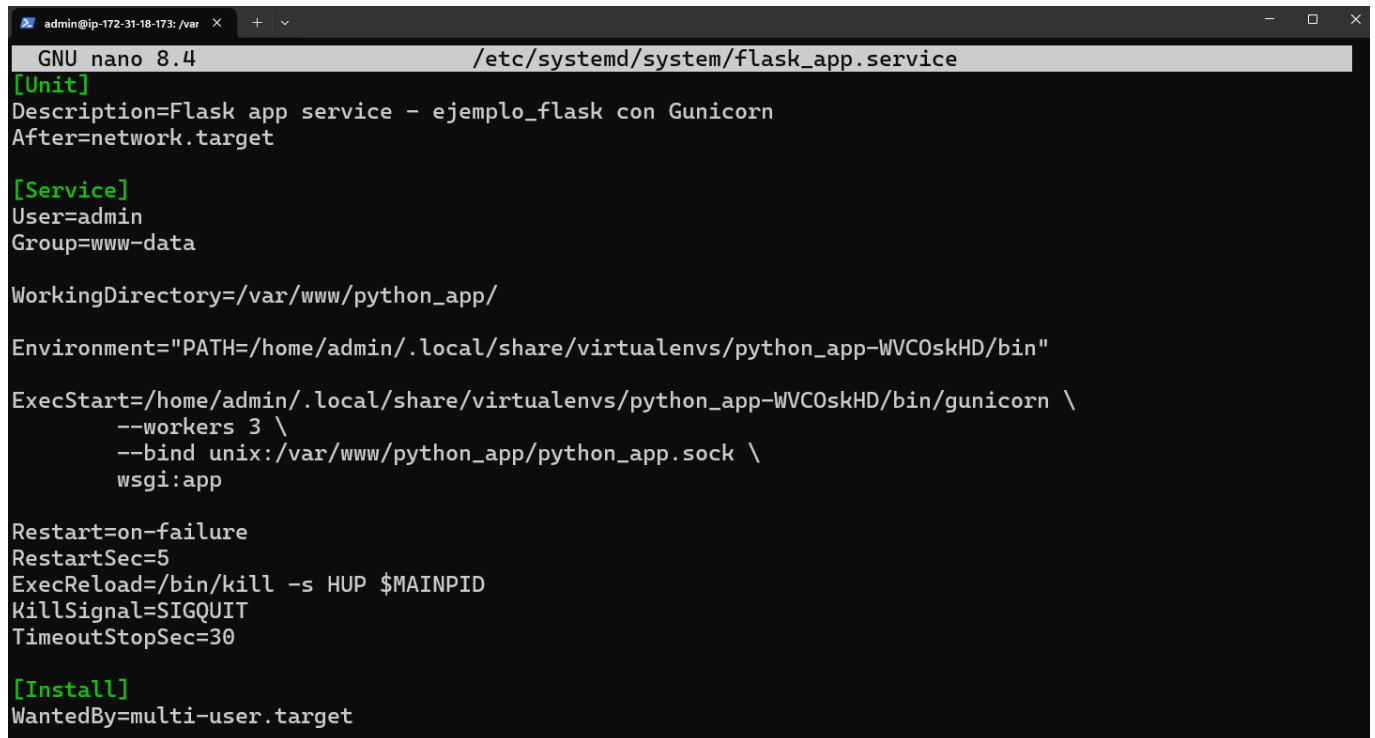
## Actividades

### ¿Qué es un unit en el contexto de Linux?

El término "unit" se refiere a cualquier tipo de recurso que coopera con Systemd, lo que incluye temporizadores, puntos de montaje, recursos de red, sockets, particiones y dispositivos sobre los servicios.

Las "configuration units", conocidas como "unit files", permiten definir cómo y cuándo se inicia un servicio, a qué recursos puede acceder y qué dependencias deben cumplirse.

### Explica cada línea del archivo de configuración del unit que has creado



```
admin@ip-172-31-18-173: /var  GNU nano 8.4 /etc/systemd/system/flask_app.service
[Unit]
Description=Flask app service - ejemplo_flask con Gunicorn
After=network.target

[Service]
User=admin
Group=www-data

WorkingDirectory=/var/www/python_app/

Environment="PATH=/home/admin/.local/share/virtualenvs/python_app-WVC0skHD/bin"

ExecStart=/home/admin/.local/share/virtualenvs/python_app-WVC0skHD/bin/gunicorn \
    --workers 3 \
    --bind unix:/var/www/python_app/python_app.sock \
    wsgi:app

Restart=on-failure
RestartSec=5
ExecReload=/bin/kill -s HUP $MAINPID
KillSignal=SIGQUIT
TimeoutStopSec=30

[Install]
WantedBy=multi-user.target
```

1. Description -> Un texto breve, significativo y legible que identifica la unidad.
2. After -> Especifica que el servicio debe iniciarse sólo después de que se haya iniciado otro servicio especificado.
3. User -> Especifica el propietario del usuario
4. Group -> Especifica el propietario del grupo
5. WorkingDirectory -> Define en qué directorio se lanzará el servicio.
6. Environment -> Apunta a las variables de entorno y las reenvía a la aplicación.
7. ExecStart -> Comandos que se ejecutan cuando se inicia este servicio.
8. Restart -> Configura si el servicio se debe reiniciar cuando el proceso de servicio finaliza, se elimina o se alcanza un tiempo de espera.
9. RestartSec -> Configura el tiempo de suspensión antes de reiniciar un servicio.
10. ExecReload -> Comandos a ejecutar para activar una recarga de configuración en el servicio.
11. KillSignal -> Especifica qué señal utilizar al detener un servicio.
12. TimeoutStopSec -> Esta opción tiene dos funciones. Primero, configura el tiempo de espera para cada comando ExecStop=. Segundo, configura el tiempo de espera para que el servicio se detenga.
13. WantedBy -> Establece el objetivo o los objetivos bajo los cuales se debe iniciar el servicio.

Busca, lee, entiende y explica qué es y para qué sirve un servidor WSGI

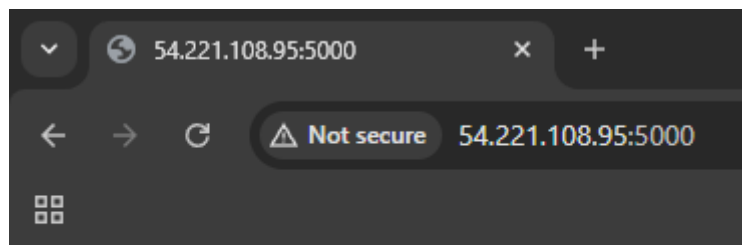
La Interfaz de Puerta de Enlace del Servidor Web (WSGI) es un mediador responsable de la comunicación entre un servidor web y una aplicación web Python. Explica cómo el servidor web se comunica con la aplicación y cómo esta puede encadenarse para procesar una solicitud.

Documenta adecuadamente con explicaciones y capturas de pantalla los procesos de despliegue de ambas aplicaciones en Flask, así como las respuestas a las cuestiones planteadas. Acuérdate de realizar el vídeo en que se muestre que el servicio creado está activo y su correcto funcionamiento.

```
admin@ip-172-31-18-173: /var
admin@ip-172-31-18-173:/var/www/python_app$ pipenv --version
^[[Apipenv, version 2024.0.1
admin@ip-172-31-18-173:/var/www/python_app$ ls -l /var/www/
total 8
drwxr-xr-x 2 root root 4096 Nov 20 22:10 html
drwxrwxr-x 3 admin www-data 4096 Nov 22 14:07 python_app
admin@ip-172-31-18-173:/var/www/python_app$ cat .env
FLASK_APP=wsgi.py
FLASK_ENV=production
admin@ip-172-31-18-173:/var/www/python_app$ cat application.py
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
    return '<h1>Aplicacion desplegada</h1>'
admin@ip-172-31-18-173:/var/www/python_app$ cat wsgi.py
from application import app

if __name__ == '__main__':
    app.run(debug=False)
admin@ip-172-31-18-173:/var/www/python_app$ pipenv shell
Loading .env environment variables...
Loading .env environment variables...
Launching subshell in virtual environment...
admin@ip-172-31-18-173:/var/www/python_app$ . /home/admin/.local/share/virtualenvs/python_app-WVC0skHD/bin/activate
(python_app) admin@ip-172-31-18-173:/var/www/python_app$ flask run --host '0.0.0.0'
* Tip: There are .env files present. Install python-dotenv to use them.
* Serving Flask app 'wsgi.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.31.18.173:5000
Press CTRL+C to quit
```



## Aplicacion desplegada

```
(python_app) admin@ip-172-31-18-173:/var/www/python_app$ sudo systemctl status flask_app.service
● flask_app.service - Flask app service - ejemplo_flask con Gunicorn
   Loaded: loaded (/etc/systemd/system/flask_app.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-11-22 15:42:11 UTC; 9s ago
     Invocation: 3054b35e715542f09a715d8f1a46a6e8
    Main PID: 987 (gunicorn)
      Tasks: 4 (limit: 4672)
     Memory: 61.9M (peak: 62.4M)
        CPU: 382ms
    CGroup: /system.slice/flask_app.service
            └─987 /home/admin/.local/share/virtualenvs/python_app-WVC0skHD/bin/python /home/admin/.local/share/virtua
              └─989 /home/admin/.local/share/virtualenvs/python_app-WVC0skHD/bin/python /home/admin/.local/share/virtua
                └─990 /home/admin/.local/share/virtualenvs/python_app-WVC0skHD/bin/python /home/admin/.local/share/virtua
                  └─991 /home/admin/.local/share/virtualenvs/python_app-WVC0skHD/bin/python /home/admin/.local/share/virtua

Nov 22 15:42:11 ip-172-31-18-173 systemd[1]: Started flask_app.service - Flask app service - ejemplo_flask con Gunicor
Nov 22 15:42:11 ip-172-31-18-173 gunicorn[987]: [2025-11-22 15:42:11 +0000] [987] [INFO] Starting gunicorn 23.0.0
Nov 22 15:42:11 ip-172-31-18-173 gunicorn[987]: [2025-11-22 15:42:11 +0000] [987] [INFO] Listening at: unix:/var/www/p
Nov 22 15:42:11 ip-172-31-18-173 gunicorn[987]: [2025-11-22 15:42:11 +0000] [987] [INFO] Using worker: sync
Nov 22 15:42:11 ip-172-31-18-173 gunicorn[989]: [2025-11-22 15:42:11 +0000] [989] [INFO] Booting worker with pid: 989
Nov 22 15:42:11 ip-172-31-18-173 gunicorn[990]: [2025-11-22 15:42:11 +0000] [990] [INFO] Booting worker with pid: 990
Nov 22 15:42:11 ip-172-31-18-173 gunicorn[991]: [2025-11-22 15:42:11 +0000] [991] [INFO] Booting worker with pid: 991
lines 1-21/21 (END)
```

```

admin@ip-172-31-18-173: /var
(python_app) admin@ip-172-31-18-173:/var/www/python_app$ cat /etc/nginx/sites-enabled/python_app.conf
server {
    listen 80;
    server_name python_app www.python_app;
    access_log /var/log/nginx/python_app.access.log;
    error_log /var/log/nginx/python_app.error.log;

    location / {
        include proxy_params;
        proxy_pass http://unix:/var/www/python_app/python_app.sock;
    }
}
(python_app) admin@ip-172-31-18-173:/var/www/python_app$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Sat 2025-11-22 14:07:03 UTC; 1h 36min ago
     Invocation: 851010c38ca041d780e65e66387265ca
       Docs: man:nginx(8)
    Process: 663 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 679 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 690 (nginx)
      Tasks: 3 (limit: 4672)
     Memory: 4.3M (peak: 4.8M)
        CPU: 33ms
    CGroup: /system.slice/nginx.service
            └─690 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              └─693 "nginx: worker process"
                └─694 "nginx: worker process"

Nov 22 14:07:03 ip-172-31-18-173 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server:
Nov 22 14:07:03 ip-172-31-18-173 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server:
lines 1-18/18 (END)

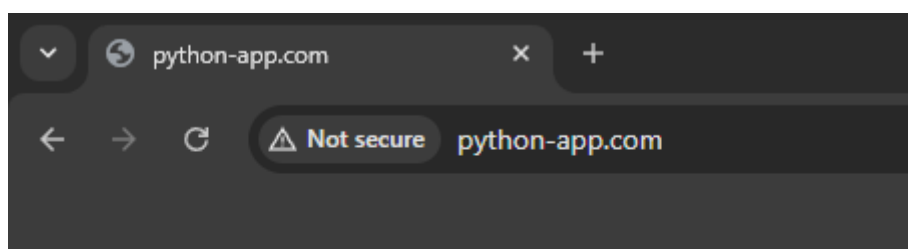
```

19

54.221.108.95

python-app

www.python-app.com



# Aplicacion desplegada