

**Yorman Leandro Trujillo Chacón - 20222209581**

## **Descargamos Python**

Se verifica que el sistema tiene las dependencias actualizadas y que no existen binarios antiguos en ejecución.

Posteriormente, se comprueba la versión instalada de Python 3, la cual será utilizada para crear el entorno virtual del proyecto.

```
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
yorman@yorman:~$ python3 --version  
Python 3.12.3  
yorman@yorman:~$
```

## **Creamos la carpeta del proyecto**

Se crea el directorio principal del proyecto dentro de la ruta /var/www/ y se asignan los permisos correspondientes al usuario actual, garantizando que tenga acceso completo a lectura y escritura.

### **Comandos utilizados:**

```
sudo mkdir -p /var/www/proyecto_basico
```

```
sudo chown -R yorman:yorman /var/www/proyecto_basico
```

```
cd /var/www/proyecto_basico
```

```
yorman@yorman:~$ sudo mkdir -p /var/www/proyecto_basico  
yorman@yorman:~$ sudo chown -R yorman:yorman /var/www/proyecto_basico  
yorman@yorman:~$ cd /var/www/proyecto_basico  
yorman@yorman:/var/www/proyecto_basico$
```

## **Crear y activar el entorno virtual**

En esta etapa se genera un entorno virtual de Python que permitirá gestionar las dependencias del proyecto sin afectar las librerías globales del sistema.

El entorno se crea con venv y luego se activa dentro del directorio del proyecto.

### **Comandos:**

```
python3 -m venv .venv
```

```
source .venv/bin/activate
```

Al activarse, el prompt de la terminal muestra el prefijo (venv) indicando que el entorno está activo.

```
yorman@yorman:/var/www/proyecto_basico$ python3 -m venv .venv
yorman@yorman:/var/www/proyecto_basico$ source .venv/bin/activate
(.venv) yorman@yorman:/var/www/proyecto_basico$
```

## Instalar Flask

Con el entorno virtual activo, se procede a instalar el framework Flask, que servirá para crear el servidor web básico.

Durante la instalación se descargan las dependencias necesarias, como Werkzeug, Jinja2 y MarkupSafe.

### Comando:

pip install flask

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ pip install --upgrade pip
Requirement already satisfied: pip in ./venv/lib/python3.12/site-packages (24.0)
Collecting pip
  Downloading pip-25.3-py3-none-any.whl.metadata (4.7 kB)
  Downloading pip-25.3-py3-none-any.whl (1.8 MB)
1.8/1.8 MB 10.7 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.0
    Uninstalling pip-24.0:
      Successfully uninstalled pip-24.0
Successfully installed pip-25.3
(.venv) yorman@yorman:/var/www/proyecto_basico$ pip install flask
pip install: command not found
(.venv) yorman@yorman:/var/www/proyecto_basico$ pip install flask
Collecting flask
  Downloading flask-3.1.2-py3-none-any.whl.metadata (3.2 kB)
Collecting blinker>=1.9.0 (from flask)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.3.0-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2.0 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting markupsafe>=2.1.1 (from flask)
  Downloading markupsafe-3.0.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (2.7 kB)
Collecting werkzeug>=3.1.0 (from flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Downloading flask-3.1.2-py3-none-any.whl (103 kB)
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.3.0-py3-none-any.whl (107 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
Downloading markupsafe-3.0.3-cp312-cp312-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (22 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Installing collected packages: markupsafe, itsdangerous, click, blinker, werkzeug, jinja2, flask
Successfully installed blinker-1.9.0 click-8.3.0 flask-3.1.2 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.3 werkzeug-3.1.3
(.venv) yorman@yorman:/var/www/proyecto_basico$
```

## Verificamos que Flask esté instalado

Se confirma la instalación correcta de Flask utilizando el siguiente comando:

pip show flask

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ pip show flask
Name: Flask
Version: 3.1.2
Summary: A simple framework for building complex web applications.
Home-page:
Author:
Author-email:
License-Expression: BSD-3-Clause
Location: /var/www/proyecto_basico/.venv/lib/python3.12/site-packages
Requires: blinker, click, itsdangerous, jinja2, markupsafe, werkzeug
Required-by:
(.venv) yorman@yorman:/var/www/proyecto_basico$
```

## Creamos el archivo app.py

Se crea el archivo principal del proyecto, donde se define la aplicación Flask. Este archivo contendrá una ruta base (/) que devuelve un mensaje en formato HTML para comprobar que el servidor funciona correctamente.

```
GNU nano 7.2 app.py *
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
    return "<h1>¡hola, Flask esta funcionando en ubuntu!</h1>"

if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=5000)
```

## Verificamos que la aplicación funcione

Ejecutamos el proyecto con:

```
python3 app.py
```

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ python3 app.py
* Serving Flask app 'app'
* Debug mode: on
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://10.0.2.15:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 121-269-801
```

## Abrimos el puerto en el navegador

Finalmente, desde el navegador accedemos a la dirección:

http://127.0.0.1:5000

El mensaje “¡Hola, Flask está funcionando en Ubuntu!” aparece correctamente, confirmando que el entorno Flask fue configurado y ejecutado con éxito.



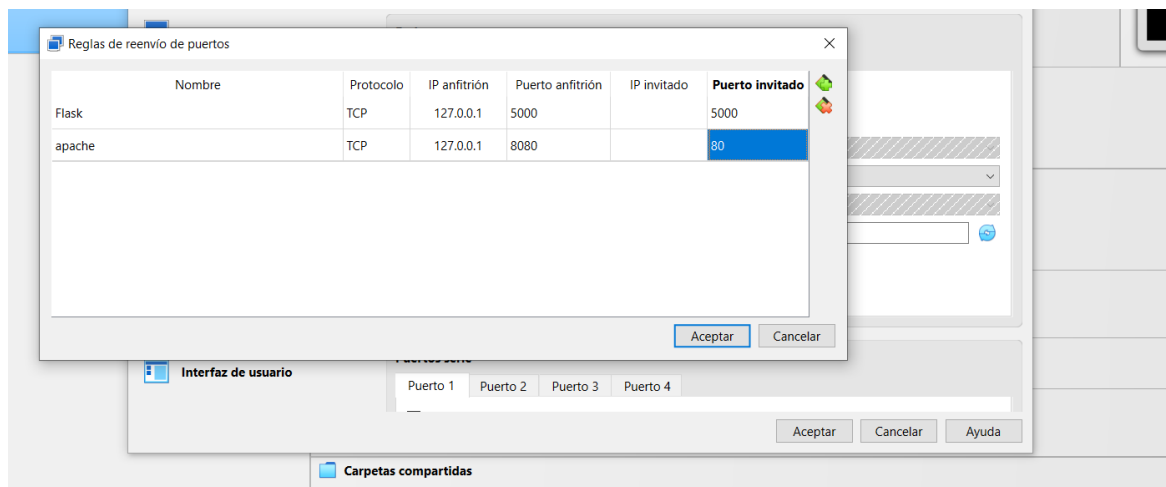
## Comenzamos configurando los puertos

En VirtualBox se configuraron las reglas de reenvío de puertos para permitir el acceso a los servicios de la máquina virtual desde el sistema anfitrión.

El puerto 5000 se asignó para Flask (modo desarrollo).

El puerto 8080 se redirigió al puerto 80, usado por Apache2.

Esto garantiza que las peticiones HTTP externas sean correctamente redirigidas al servidor web dentro del entorno Linux.



## Creamos requirements.txt e instalamos las dependencias

Se creó un archivo requirements.txt con las librerías necesarias para el proyecto Flask.

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ pip install -r requirements.txt
Requirement already satisfied: flask==3.1.2 in ./venv/lib/python3.12/site-packages (from -r requirements.txt (line 1)) (3.1.2)
Collecting gunicorn==21.2.0 (from -r requirements.txt (line 2))
  Downloading gunicorn-21.2.0-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: blinker>=1.9.0 in ./venv/lib/python3.12/site-packages (from flask==3.1.2->-r requirements.txt (line 1)) (1.9.0)
Requirement already satisfied: click>=8.1.3 in ./venv/lib/python3.12/site-packages (from flask==3.1.2->-r requirements.txt (line 1)) (8.3.0)
Requirement already satisfied: itsdangerous>=2.2.0 in ./venv/lib/python3.12/site-packages (from flask==3.1.2->-r requirements.txt (line 1)) (2.2.0)
Requirement already satisfied: Jinja2>=3.1.2 in ./venv/lib/python3.12/site-packages (from flask==3.1.2->-r requirements.txt (line 1)) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in ./venv/lib/python3.12/site-packages (from flask==3.1.2->-r requirements.txt (line 1)) (3.0.3)
Requirement already satisfied: Werkzeug>=3.1.0 in ./venv/lib/python3.12/site-packages (from flask==3.1.2->-r requirements.txt (line 1)) (3.1.3)
Collecting packaging (from gunicorn==21.2.0->-r requirements.txt (line 2))
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Downloading gunicorn-21.2.0-py3-none-any.whl (80 kB)
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Installing collected packages: packaging, gunicorn
Successfully installed gunicorn-21.2.0 packaging-25.0
(.venv) yorman@yorman:/var/www/proyecto_basico$
```

## Creamos wsgi.py y verificamos que funcione

El archivo wsgi.py actúa como punto de entrada entre Flask y Gunicorn.

Su contenido fue probado con el siguiente comando para confirmar que la aplicación Flask respondía correctamente:

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ python -c "from wsgi import app; print('OK WSGI')"
```

OK WSGI

```
(.venv) yorman@yorman:/var/www/proyecto_basico$
```

## Crear el archivo del servicio de Gunicorn

Se creó un archivo de configuración para Gunicorn en la ruta:

/etc/systemd/system/gunicorn\_proyecto.service

Este archivo permite ejecutar automáticamente Gunicorn al iniciar el sistema.

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo systemctl daemon-reload
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo systemctl restart gunicorn_proyecto
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo systemctl status gunicorn_proyecto --no-pager
● gunicorn_proyecto.service - Gunicorn proyecto_basico
   Loaded: loaded (/etc/systemd/system/gunicorn_proyecto.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-11-05 21:54:44 UTC; 4s ago
     Main PID: 1344 (gunicorn)
       Tasks: 3 (limit: 2268)
      Memory: 44.5M (peak: 44.7M)
         CPU: 617ms
    CGroup: /system.slice/gunicorn_proyecto.service
            └─1344 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app
              └─1345 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app
              └─1346 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app

nov 05 21:54:44 yorman systemd[1]: Started gunicorn_proyecto.service - Gunicorn proyecto_basico.
nov 05 21:54:44 yorman gunicorn[1344]: [2025-11-05 21:54:44 +0000] [1344] [INFO] Starting gunicorn 21.2.0
nov 05 21:54:44 yorman gunicorn[1344]: [2025-11-05 21:54:44 +0000] [1344] [INFO] Listening at: http://127.0.0.1:8000 (1344)
nov 05 21:54:44 yorman gunicorn[1344]: [2025-11-05 21:54:44 +0000] [1344] [INFO] Using worker: sync
nov 05 21:54:44 yorman gunicorn[1345]: [2025-11-05 21:54:44 +0000] [1345] [INFO] Booting worker with pid: 1345
nov 05 21:54:44 yorman gunicorn[1346]: [2025-11-05 21:54:44 +0000] [1346] [INFO] Booting worker with pid: 1346
nov 05 21:54:49 yorman systemd[1]: /etc/systemd/system/gunicorn_proyecto.service, ignoring line.
Hint: Some lines were ellipsized, use -l to show in full.
(.venv) yorman@yorman:/var/www/proyecto_basico$
```

## Instalar Apache y módulos de proxy

Se instaló Apache2 y se habilitaron los módulos necesarios para configurar el proxy inverso.

Esto permite que Apache redirija las solicitudes HTTP al puerto donde corre Gunicorn (8000).

### Comando utilizado:

```
sudo a2enmod proxy proxy_http
```

```
Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo a2enmod proxy proxy_http
Enabling module proxy.
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

## Crear el sitio proyecto\_basico.conf

Se creó el archivo de configuración del sitio en:

/etc/apache2/sites-available/proyecto\_basico.conf

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ curl -I http://127.0.0.1/
HTTP/1.1 200 OK
Date: Wed, 05 Nov 2025 23:22:16 GMT
Server: gunicorn
Content-Type: text/html; charset=utf-8
Content-Length: 50
```

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress th
is message
Syntax OK
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo systemctl restart apache2
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo systemctl status gunicorn_projecto --no-pager
● gunicorn_projecto.service - Gunicorn proyecto_basico
   Loaded: loaded (/etc/systemd/system/gunicorn_projecto.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-11-05 21:54:44 UTC; 1h 22min ago
     Main PID: 1344 (gunicorn)
        Tasks: 3 (limit: 2268)
      Memory: 44.5M (peak: 44.7M)
         CPU: 1.830s
    CGroup: /system.slice/gunicorn_projecto.service
            └─1344 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app
              └─1345 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app
                └─1346 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app

nov 05 21:54:44 yorman gunicorn[1345]: [2025-11-05 21:54:44 +0000] [1345] [INFO] Booting worker with pid: 1345
nov 05 21:54:44 yorman gunicorn[1346]: [2025-11-05 21:54:44 +0000] [1346] [INFO] Booting worker with pid: 1346
nov 05 21:54:49 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 22:11:24 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 22:11:24 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 22:11:25 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 22:11:26 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 22:11:26 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 22:11:26 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 22:11:27 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
nov 05 23:16:52 yorman systemd[1]: /etc/systemd/system/gunicorn_projecto.service.gnoring line.
Hint: Some lines were ellipsized, use -l to show in full.
```

## Crear el archivo del sitio proyecto\_basico.conf

En esta etapa se configura el archivo del sitio dentro de la **ruta**:

/etc/apache2/sites-available/proyecto\_basico.conf

Luego, se activó el sitio con:

```
sudo a2ensite proyecto_basico.conf
```

```
sudo systemctl restart apache2
```

Esto permitió establecer la conexión entre Apache (puerto 80) y Gunicorn (puerto 8000).



## Estructura final del proyecto

Después de crear todos los archivos y carpetas necesarias mediante Nano, la estructura final del proyecto quedó así:

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ rm -f app.py requirements.txt
(.venv) yorman@yorman:/var/www/proyecto_basico$ ls -l
total 16
drwxr-xr-x 5 yorman yorman 4096 nov  5 23:56 app
drwxrwxr-x 2 yorman yorman 4096 nov  5 19:46 __pycache__
-rw-rw-r-- 1 yorman yorman  30 nov  5 19:34 requirements.txt
-rw-rw-r-- 1 yorman yorman  59 nov  5 19:44 wsgi.py
(.venv) yorman@yorman:/var/www/proyecto_basico$
```

# Vamos a correr el proyecto

## Reiniciamos Gunicorn

Para que Gunicorn cargue la nueva estructura del proyecto y los archivos actualizados, se ejecutan los siguientes **comandos**:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart gunicorn_proyecto
```

```
sudo systemctl status gunicorn_proyecto --no-pager
```

El estado active (running) confirma que el servicio se encuentra funcionando correctamente.

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo systemctl restart gunicorn_proyecto
[sudo] password for yorman:
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo status gunicorn_proyecto --no-pager
sudo: status: command not found
(.venv) yorman@yorman:/var/www/proyecto_basico$ sudo systemctl status gunicorn_proyecto --no-pager
● gunicorn_proyecto.service - Gunicorn proyecto_basico
   Loaded: loaded (/etc/systemd/system/gunicorn_proyecto.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-11-06 00:25:53 UTC; 58s ago
     Main PID: 3039 (gunicorn)
       Tasks: 3 (limit: 2268)
      Memory: 40.9M (peak: 41.1M)
         CPU: 478ms
    CGroup: /system.slice/gunicorn_proyecto.service
            └─3039 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app
            └─3042 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app
            └─3043 /var/www/proyecto_basico/.venv/bin/python3 /var/www/proyecto_basico/.venv/bin/gunicorn --workers 2 --bind 127.0.0.1:8000 wsgi:app

nov 06 00:25:53 yorman systemd[1]: Started gunicorn_proyecto.service - Gunicorn proyecto_basico.
nov 06 00:25:53 yorman gunicorn[3039]: [2025-11-06 00:25:53 +0000] [3039] [INFO] Starting gunicorn 21.2.0
nov 06 00:25:53 yorman gunicorn[3039]: [2025-11-06 00:25:53 +0000] [3039] [INFO] Listening at: http://127.0.0.1:8000 (3039)
nov 06 00:25:53 yorman gunicorn[3039]: [2025-11-06 00:25:53 +0000] [3039] [INFO] Using worker: sync
nov 06 00:25:53 yorman gunicorn[3042]: [2025-11-06 00:25:53 +0000] [3042] [INFO] Booting worker with pid: 3042
nov 06 00:25:53 yorman gunicorn[3043]: [2025-11-06 00:25:53 +0000] [3043] [INFO] Booting worker with pid: 3043
```

## Probamos si Flask responde

Para verificar la respuesta del servidor Gunicorn, se ejecuta una prueba local con **curl**:

`curl -I http://127.0.0.1:8000/`

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ curl -I http://127.0.0.1:8000/
HTTP/1.1 200 OK
Server: gunicorn
Date: Thu, 06 Nov 2025 00:29:23 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 297
```

## Verificación del proxy inverso (Apache)

A continuación, se valida la conexión del proxy inverso de Apache hacia Gunicorn mediante el mismo comando **curl**:

`curl -I http://127.0.0.1/`

El resultado muestra nuevamente un estado 200 OK, confirmando que Apache reenvía correctamente las solicitudes HTTP hacia Gunicorn.

```
(.venv) yorman@yorman:/var/www/proyecto_basico$ curl -I http://127.0.0.1/
HTTP/1.1 200 OK
Date: Thu, 06 Nov 2025 00:39:48 GMT
Server: gunicorn
Content-Type: text/html; charset=utf-8
Content-Length: 297
```

Prueba final en el navegador

Finalmente, desde el sistema anfitrión se accede a la dirección:

`http://127.0.0.1:8080`

**El navegador muestra la página con el mensaje:**



¡Funciona vía Apache + Gunicorn!

Servidor listo para la demo. Estructura tipo paquete Flask.

Esto confirma que la aplicación Flask está funcionando correctamente a través de Gunicorn y Apache2, cumpliendo con el objetivo del proyecto.

