

## Despliegue de aplicación calculadora

# Guía de despliegue en servidor Ubuntu 20

[illegible]

## Contenidos

1. [¿Qué es esta documentación?]
2. [Instalación de paquetes ubuntu]
3. [Obtener los archivos fuente]
4. [Preparar aplicación Flask]
  1. [Instalación de módulos de python3]
  2. [Ejecución del Flask http server]
5. [Preparar Apache como proxy]
  1. [Activar módulos de apache]
  2. [Configuración de virtualhost de apache2]
  3. [Probar Proxy Apache]
6. [Resolución de problemas]

## ¿Qué es esta documentación?

Pasos a seguir para desplegar una aplicación web Flask que se sirve por apache a través de un proxy.

Esta documentación también se puede leer a través de GitHub (se recomienda).

(Guía deploy Github)[<https://github.com/daniel-domínguez-daw/m8-uf2-proyecto/blob/main/docs/deploy.md>]

## Instalación de paquetes ubuntu

Partimos de un sistema Ubuntu 20.04. Se deben instalar los siguientes paquetes en el servidor.

```
sudo apt update
```

- (Opcional) Git: `sudo apt install git`
- Apache2: `sudo apt install apache2`
- Python3: `sudo apt install python3`

## Obtener los archivos fuente

(Opcional) Si hemos decidido usar Git podemos traer el código fuente tanto de la aplicación cliente como de la aplicación servidor que están alojadas en el repositorio github `https://github.com/daniel-domínguez-daw/m8-uf2-proyecto/` , sino podemos utilizar un simple unzip.

```
https://github.com/daniel-domínguez-daw/m8-uf2-proyecto/archive/main.zip
```

Usando git o descargando los sources desde el archivo zip dejaremos los archivos bajo la carpeta home por ejemplo

```
~/calc/src/ .
```

## Preparar aplicación Flask

### Instalación de módulos de python3

Se debe crear un entorno virtual de python3 donde instalaremos los módulos necesarios para hacer funcionar la API Flask.

```
cd ~/calc/src
python3 -m venv env
source env/bin/activate
```

A continuación hay que instalar las dependencias de python.

```
(env) $ pip3 install -r requirements.txt
```

## Ejecución del Flask http server

Ejecución en segundo plano.

```
cd ~/calc/src  
python3 calc.py &
```

Probamos con curl:

```
curl localhost:5050/suma/2/3/  
# {'result': 5}
```

Ahora que funciona la aplicación Flask se debe configurar apache para servir esto desde el 8080.

## Preparar Apache como proxy

### Activar módulos de apache

Para redirigir el tráfico de apache a Flask se deben activar los mods `proxy` y `proxy_http`.

```
$ sudo a2enmod proxy  
$ sudo a2enmod proxy_http  
$ sudo systemctl restart apache2
```

### Configuración de virtualhost de apache2

Como apache no va a servir ningún archivo estático salvo el error 503, esta es la configuración de virtualhost necesaria.

```
<VirtualHost *:8080>  
    ServerName calc  
    ProxyPass / http://localhost:5050/  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
    ErrorDocument 503 /var/www/error/503.html  
</VirtualHost>
```

También se puede encontrar en la raíz del repositorio `etc/calc.conf`.

Recordar que si se copia tal cual en la carpeta de apache sites-available se ha de hacer un `a2ensite calc`.  
Recordar también que solo puede haber una configuración con el puerto 8080 (salvo que se utilicen nombres de dominio) por lo tanto quizás hay que desactivar otras configuraciones o elegir otro puerto diferente.

Creamos el archivo html de error 503 para cuando el proxy esté en problemas.

```
mkdir /var/www/error  
touch /var/www/error/503.html  
vim /var/www/error/503.html
```

El contenido del archivo `503.html` debe ser similar al mostrado a continuación

```
<html>
  <head>
    <title>Error 503 Service Unavailable</title>
  </head>
  <body>
    <h1>Tenemos problemas</h1>
    <p>El sitio web está en problemas. Si persiste contacte con admin@daw2m8.org</p>
  </body>
</html>
```

## Probar Proxy Apache

Simplemente visitamos <http://localhost:8080/> desde un navegador si utilizamos ubuntu, sino podemos usar el navegador de terminal links o curl.

## Resolución de problemas

No debería ocurrir pero en caso de que se intente acceder a una ruta que no está contemplada en la aplicación Flask el servidor devolverá un error de cliente 404 Not Found.

En el caso de que Flask caiga o diera algún tipo de error de servidor (que en principio no debería), devolverá un error 503 el cual apache2 nos ofrecerá una página estática de error. Este error nos aparecerá también en los logs de apache.

En caso de error 503, se debe comprobar si la aplicación Flask está caída. Para ello podemos utilizar el comando `ps aux | grep flask`. Nos listará el listado de procesos por si queremos además realizar un `kill` pasándole el id del proceso.

En caso que esté caído, para solucionarlo simplemente hay que volver a levantar Flask.

```
cd ~/calc/src
python3 calc.py &
```