


Despliegue de aplicación Django en PythonAnywhere

Configuraciones para instalar un proyecto Django 3.2 en PythonAnywhere.

Tabla de contenidos

Despliegue de aplicación Django en PythonAnywhere.....	1
Información inicial.....	2
Pasos para Despliegue 	3
1. Crear y activar entorno virtual para python.....	4
Crear un entorno virtual.....	4
Desactivar y reactivar el virtualenv.....	6
Instalar paquetes en su virtualenv.....	7
Configurar la aplicación para usar este entorno virtual.....	7
2. Crear nueva web app en pythonanywhere.....	8
Paso 1: Crear nueva web app.....	8
Paso 2: Crear nuevo framework de python.....	9
Paso 3: Seleccionar versión de python.....	10
Paso 4: Seleccionar virtualenv.....	11
3. Configuración de webapp en servidor.....	12
Paso 1: Clonar repositorio privado de github.....	12
Paso 2: Instalar los requerimientos.....	13
4. Configurar static files.....	14
Paso 1: Acceder a la pestaña web.....	14
Paso 2. Bajar hasta la sección "Static files".....	14
Paso 3. Configurar rutas estáticas en settings.py.....	15
Paso 4. Recolectar estáticos.....	15
5. Modificar WSGI.....	16
Paso 1. Acceder a la pestaña web.....	16
Paso 2. Bajar hasta la sección "Code".....	16
Paso 3. Editar archivo de configuración wsgi.....	17
6. Otras configuraciones.....	19
7. Otra info.....	19

Información inicial

Lo primero a tener en cuenta si estás usando una cuenta gratuita, es tener paciencia. Estas instrucciones sirven para la instalación de un blog creado en Django usando un diseño personalizado.

Se usaron las siguientes versiones base en local:

- **Lenguaje de programación:** Python 3.6.9
- **Gestor de paquetes:** Pip 9.0.1
- **Framework web:** Django 3.2.16
- **Base de datos:** Sqlite3

Generar token en github para el servidor

- <https://github.com/settings/tokens>
- ghp_TOKEN_GENERADO_EN_GITHUB

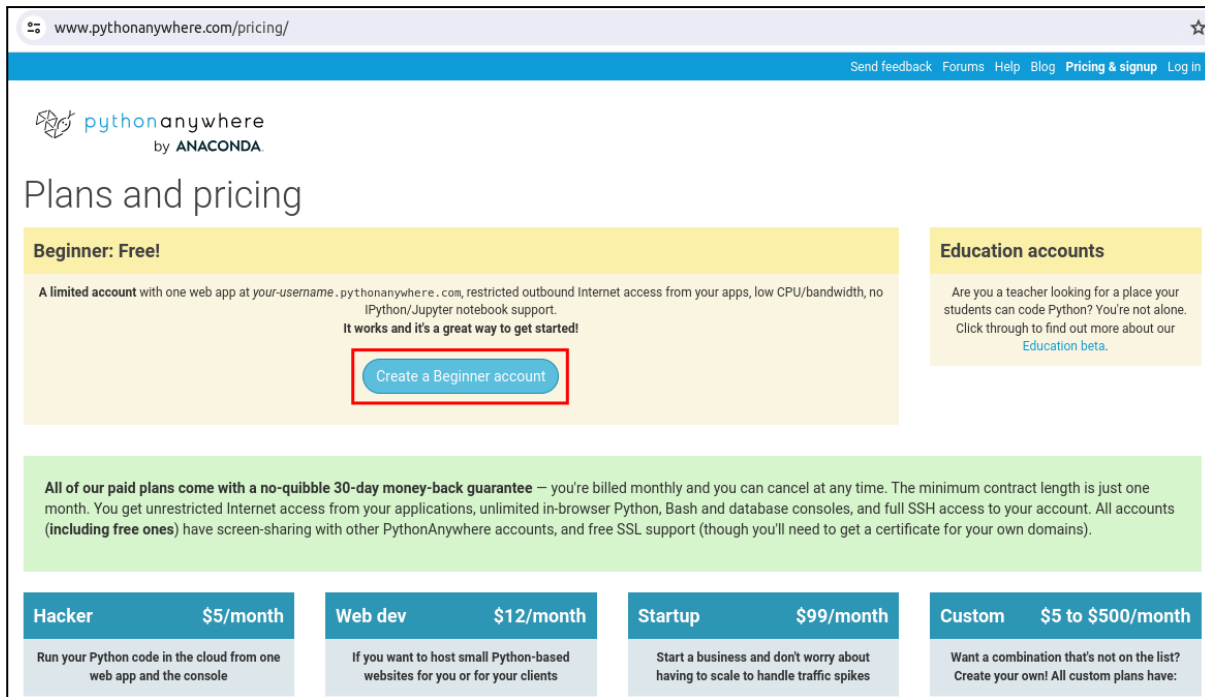
Ayuda

- **Deploying an existing Django project on PythonAnywhere:**
 - <https://help.pythonanywhere.com/pages/DeployExistingDjangoProject/>
- **Deploy a Django web app to Python Anywhere [FREE]:**
 - <https://www.youtube.com/watch?v=xtnUwvjOThg>
- **How to Deploy Django Web App to Pythonanywhere (with static files and media):**
 - <https://www.youtube.com/watch?v=Gnwm7fQnt2c>

Pasos para Despliegue 🚀

Se asume que existe previamente una cuenta beginner en pythonanywhere:

<https://www.pythonanywhere.com/pricing/>



www.pythonanywhere.com/pricing/

Send feedback Forums Help Blog Pricing & signup Log in

pythonanywhere by ANACONDA

Plans and pricing

Beginner: Free!

A limited account with one web app at `your-username.pythonanywhere.com`, restricted outbound Internet access from your apps, low CPU/bandwidth, no IPython/Jupyter notebook support.
It works and it's a great way to get started!

Create a Beginner account

Education accounts

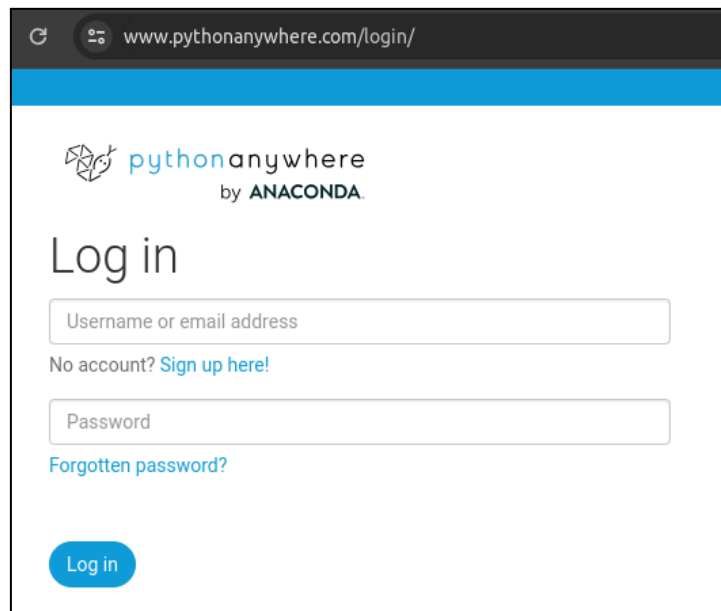
Are you a teacher looking for a place your students can code Python? You're not alone. Click through to find out more about our [Education beta](#).

All of our paid plans come with a no-quibble 30-day money-back guarantee — you're billed monthly and you can cancel at any time. The minimum contract length is just one month. You get unrestricted Internet access from your applications, unlimited in-browser Python, Bash and database consoles, and full SSH access to your account. All accounts (including free ones) have screen-sharing with other PythonAnywhere accounts, and free SSL support (though you'll need to get a certificate for your own domains).

Hacker	\$5/month	Web dev	\$12/month	Startup	\$99/month	Custom	\$5 to \$500/month
Run your Python code in the cloud from one web app and the console		If you want to host small Python-based websites for you or for your clients		Start a business and don't worry about having to scale to handle traffic spikes		Want a combination that's not on the list? Create your own! All custom plans have:	

Acceder a pythonanywhere

- <https://www.pythonanywhere.com/user/xqazprog/>
- Agregar usuario y password



www.pythonanywhere.com/login/

pythonanywhere by ANACONDA

Log in

Username or email address

No account? [Sign up here!](#)

Password

[Forgotten password?](#)

Log in

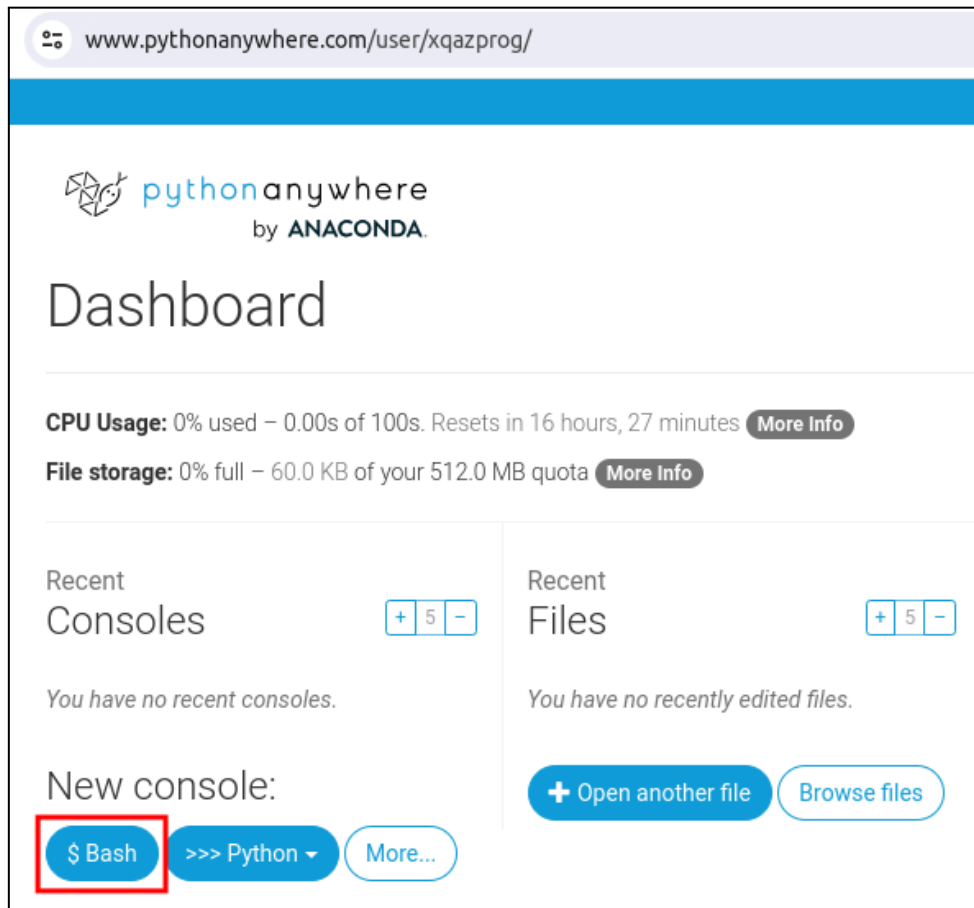
1. Crear y activar entorno virtual para python

- Versiones de python se ubican en `/usr/bin/`

Puede usar un virtualenv en una nueva aplicación web (creada usando la opción "Configuración manual") o en sus aplicaciones web existentes. Para utilizar un virtualenv en su aplicación web, haga lo siguiente.

Crear un entorno virtual

Vaya a la pestaña **Consoles** e inicie una consola **Bash**.

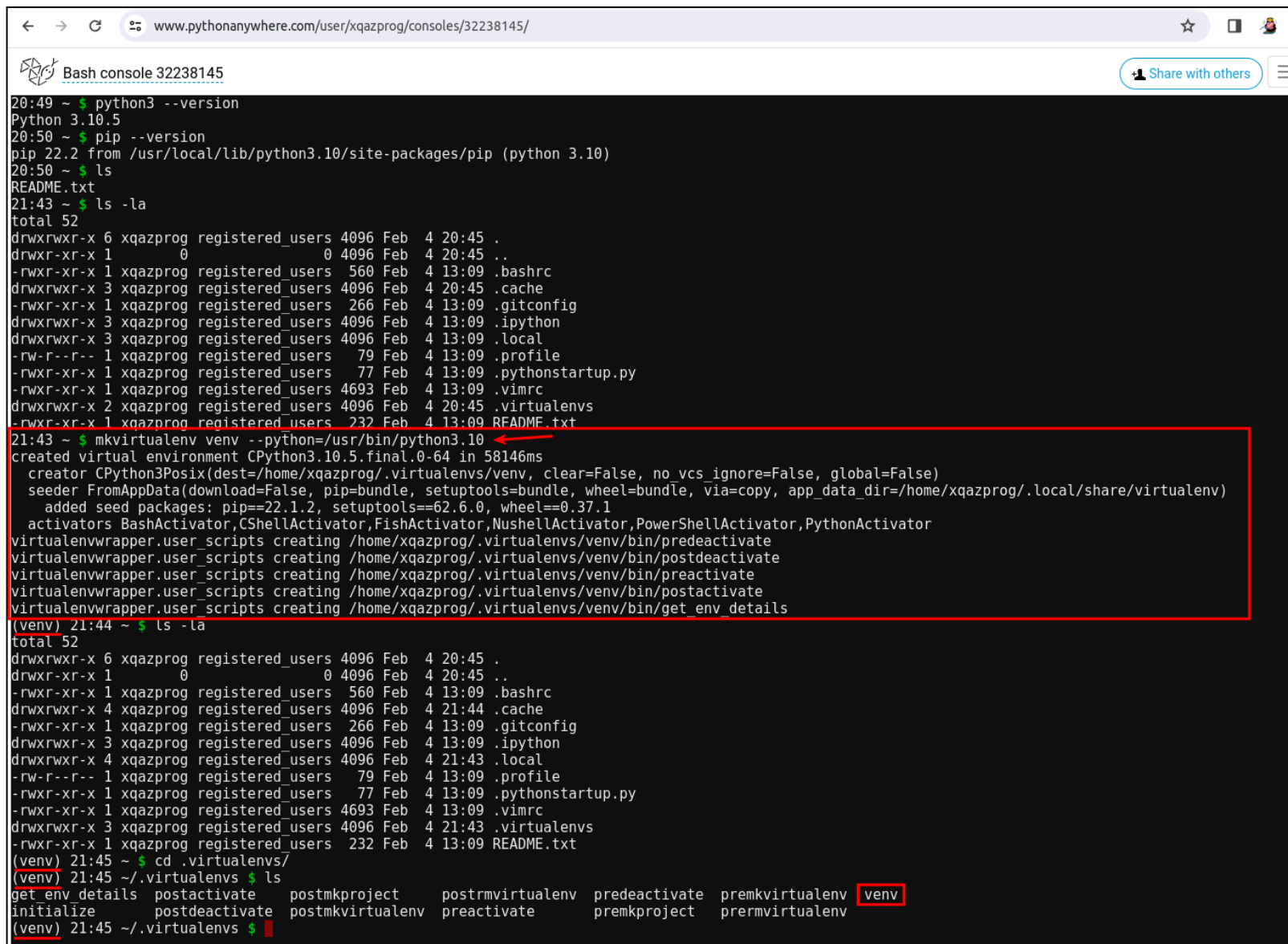


Recomendamos utilizar `virtualenvwrapper`, una práctica herramienta de línea de comandos, para crear su `virtualenv`.

Especifica qué versión de Python usar para tu `virtualenv` usando la opción `--python`, pero ten en cuenta que debe coincidir con la versión de Python que has elegido para tu aplicación web. Entonces, para crear un nuevo entorno virtual Python 3.10, ejecute este comando:

- `mkvirtualenv venv --python=/usr/bin/python3.6`
- `mkvirtualenv venv --python=/usr/bin/python3.10`

En la flecha se puede ver cómo se crea el entorno virtual.



```
20:49 ~ $ python3 --version
Python 3.10.5
20:50 ~ $ pip --version
pip 22.2 from /usr/local/lib/python3.10/site-packages/pip (python 3.10)
20:50 ~ $ ls
README.txt
21:43 ~ $ ls -la
total 52
drwxrwxr-x 6 xqazprog registered_users 4096 Feb  4 20:45 .
drwxr-xr-x 1 0 0 4096 Feb  4 20:45 ..
-rwxr-xr-x 1 xqazprog registered_users 560 Feb  4 13:09 .bashrc
drwxrwxr-x 3 xqazprog registered_users 4096 Feb  4 20:45 .cache
-rwxr-xr-x 1 xqazprog registered_users 266 Feb  4 13:09 .gitconfig
drwxrwxr-x 3 xqazprog registered_users 4096 Feb  4 13:09 .ipython
drwxrwxr-x 3 xqazprog registered_users 4096 Feb  4 13:09 .local
-rw-r--r-- 1 xqazprog registered_users 79 Feb  4 13:09 .profile
-rwxr-xr-x 1 xqazprog registered_users 77 Feb  4 13:09 .pythonstartup.py
-rwxr-xr-x 1 xqazprog registered_users 4693 Feb  4 13:09 .vimrc
drwxrwxr-x 2 xqazprog registered_users 4096 Feb  4 20:45 .virtualenvs
-rwxr-xr-x 1 xqazprog registered_users 232 Feb  4 13:09 README.txt
21:43 ~ $ mkvirtualenv venv --python=/usr/bin/python3.10
created virtual environment CPython3.10.5.final.0-64 in 58146ms
  creator CPython3Posix(dest=/home/xqazprog/.virtualenvs/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/xqazprog/.local/share/virtualenv)
    added seed packages: pip==22.1.2, setuptools==62.6.0, wheel==0.37.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /home/xqazprog/.virtualenvs/venv/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/xqazprog/.virtualenvs/venv/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/xqazprog/.virtualenvs/venv/bin/preactivate
virtualenvwrapper.user_scripts creating /home/xqazprog/.virtualenvs/venv/bin/postactivate
virtualenvwrapper.user_scripts creating /home/xqazprog/.virtualenvs/venv/bin/get_env_details
(venv) 21:44 ~ $ ls -la
total 52
drwxrwxr-x 6 xqazprog registered_users 4096 Feb  4 20:45 .
drwxr-xr-x 1 0 0 4096 Feb  4 20:45 ..
-rwxr-xr-x 1 xqazprog registered_users 560 Feb  4 13:09 .bashrc
drwxrwxr-x 4 xqazprog registered_users 4096 Feb  4 21:44 .cache
-rwxr-xr-x 1 xqazprog registered_users 266 Feb  4 13:09 .gitconfig
drwxrwxr-x 3 xqazprog registered_users 4096 Feb  4 13:09 .ipython
drwxrwxr-x 4 xqazprog registered_users 4096 Feb  4 21:43 .local
-rw-r--r-- 1 xqazprog registered_users 79 Feb  4 13:09 .profile
-rwxr-xr-x 1 xqazprog registered_users 77 Feb  4 13:09 .pythonstartup.py
-rwxr-xr-x 1 xqazprog registered_users 4693 Feb  4 13:09 .vimrc
drwxrwxr-x 3 xqazprog registered_users 4096 Feb  4 21:43 .virtualenvs
-rwxr-xr-x 1 xqazprog registered_users 232 Feb  4 13:09 README.txt
(venv) 21:45 ~ $ cd .virtualenvs/
(venv) 21:45 ~/.virtualenvs $ ls
get_env_details  postactivate  postmkproject  postrmvirtualenv  predeactivate  premkvirtualenv  venv
initialize      postdeactivate  postmkvirtualenv  preactivate      premkproject   prermvirtualenv
(venv) 21:45 ~/.virtualenvs $
```

Una vez que su virtualenv esté listo y activo, verá `(env)` \$ en su mensaje.

Desactivar y reactivar el virtualenv

Una vez que cree su virtualenv, debe activarlo. Se activa automáticamente inmediatamente después de crearlo con `mkvirtualenv`. Se puede activar usando `workon` y para desactivar, utilice `deactivate`:

```
Bash console 32238145
22:45 ~ $ workon venv
(venv) 22:46 ~ $ which python
/home/xqazprog/.virtualenvs/venv/bin/python
(venv) 22:46 ~ $ deactivate
22:46 ~ $ which python
/home/xqazprog/.local/bin/python
22:46 ~ $
```

También se puede activar usando:

- `source /home/usuario/.virtualenvs/venv/bin/activate`

```
Bash console 32238145
23:02 ~ $ source /home/xqazprog/.virtualenvs/venv/bin/activate
(venv) 23:02 ~ $ which python
/home/xqazprog/.virtualenvs/venv/bin/python
(venv) 23:02 ~ $
```

También se puede generar una nueva consola bash con el entorno virtual activado desde la pestaña web.

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[/home/xqazprog/.virtualenvs/venv](#)

[Start a console in this virtualenv](#)

```
← → ↻ www.pythonanywhere.com/user/xqazprog/consoles/32238145/

Bash console 32238145
(venv) 22:35 ~ $ which python
/home/xqazprog/.virtualenvs/venv/bin/python
(venv) 22:35 ~ $ which pip
/home/xqazprog/.virtualenvs/venv/bin/pip
(venv) 22:36 ~ $ python --version
Python 3.10.5
(venv) 22:37 ~ $ python3 --version
Python 3.10.5
(venv) 22:37 ~ $ pip --version
pip 22.1.2 from /home/xqazprog/.virtualenvs/venv/lib/python3.10/site-packages/pip (python 3.10)
(venv) 22:37 ~ $
```

Instalar paquetes en su virtualenv

Para instalar paquetes en el virtualenv se puede utilizar pip

- `pip3 install django`
- `pip3 install whitenoise`

Para instalar una versión específica de Django

- `pip3 install django==3.2.6`

Configurar la aplicación para usar este entorno virtual

Ahora que tiene un entorno virtual y conoce su ruta, configure su aplicación para usar este entorno virtual.

Vaya a la pestaña Web y en la sección Virtualenv, ingrese la ruta:

- `/home/nombreusuario/.virtualenvs/myvirtualenv`
- `/home/xqazprog/.virtualenvs/venv`

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

`/home/xqazprog/.virtualenvs/venv`

[Start a console in this virtualenv](#)



SUGERENCIA

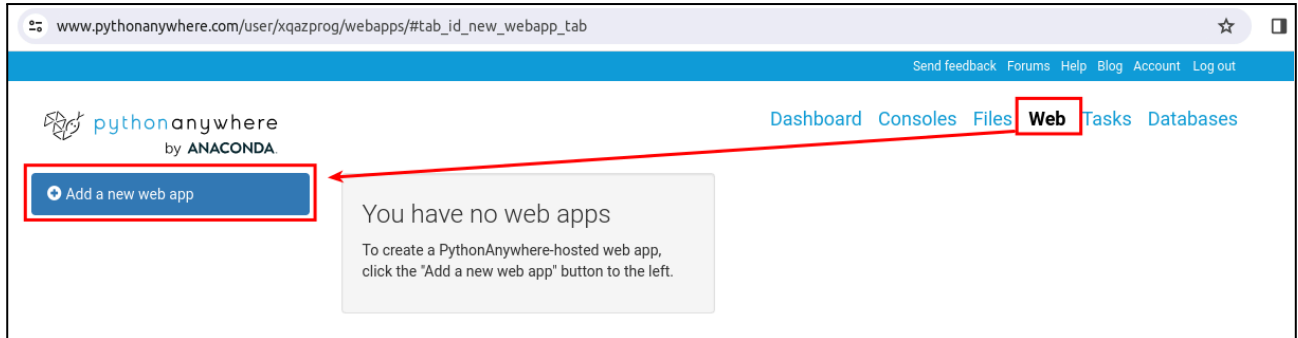
Si está utilizando virtualenvwrapper, puede simplemente ingresar el nombre del virtualenv, myvirtualenv, y el sistema adivinará automáticamente el resto de la ruta (/home/myusername/.virtualenvs, etc.) después de presionar ok.

Ahora, vuelva a cargar su aplicación web y debería encontrar que tiene acceso a todos los paquetes en su virtualenv, en lugar de tener accesos a los paquetes del sistema.

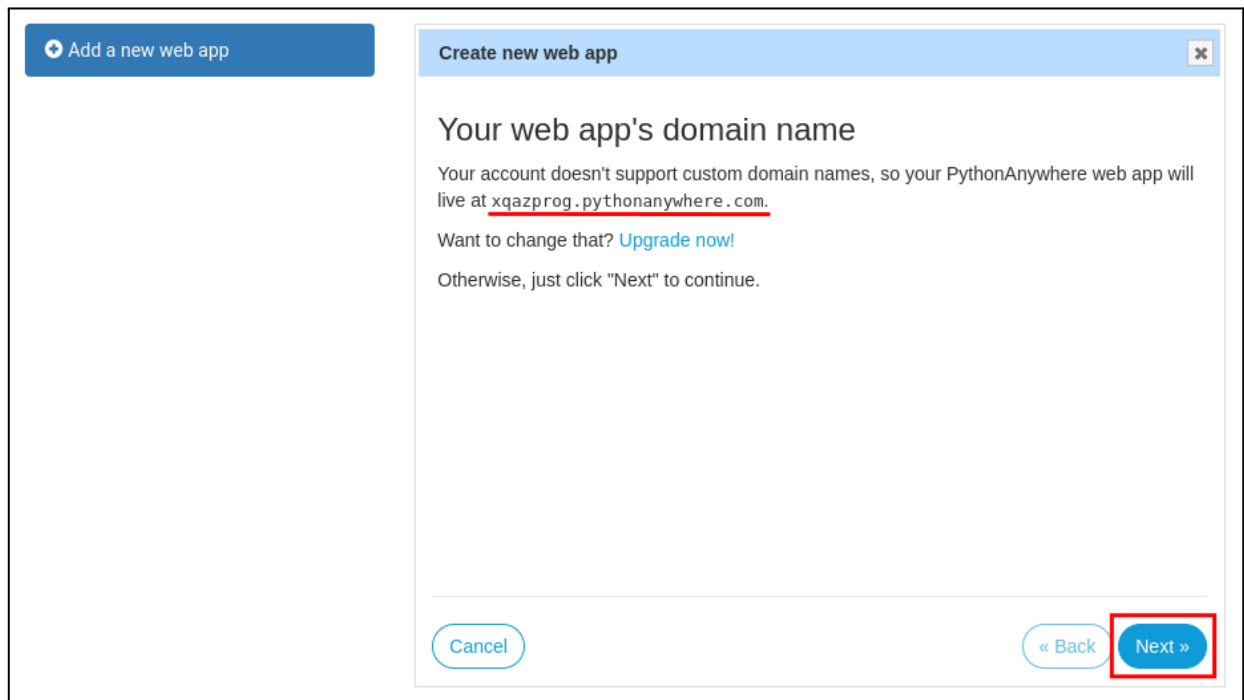
2. Crear nueva web app en pythonanywhere

Paso 1: Crear nueva web app

- <https://www.pythonanywhere.com/user/xqazprog/webapps/>
- **Nombre:** xqazprog.pythonanywhere.com



Luego de presionar el botón, se deben seguir las instrucciones para crear un nuevo nombre de dominio para la aplicación.



Paso 2: Crear nuevo framework de python

- Escoger configuración manual
 - La configuración manual implica editar su propio archivo de configuración WSGI en `/var/www/`. Por lo general, esto importa una aplicación compatible con WSGI que ha almacenado en otro lugar
 - Cuando haga clic en "Siguiente", crearemos un archivo WSGI para usted, incluida una aplicación simple "Hello World" que puede usar para comenzar, así como algunos comentarios sobre cómo usar otros marcos.
 - También podrá especificar un virtualenv para usar en su aplicación.

Create new web app

Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » Django
- » web2py
- » Flask
- » Bottle
- » **Manual configuration (including virtualenvs)**

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Cancel

« Back

Next »

Paso 3: Seleccionar versión de python

- Versiones desde 3.6 a 3.10
- Escoger Python 3.6

Create new web app

Select a Python version

» Python 2.7

» Python 3.6

» Python 3.7

» Python 3.8

» Python 3.9

» Python 3.10

Cancel

« Back

Next »

Create new web app

Manual Configuration

Manual configuration involves editing your own WSGI configuration file in `/var/www/`. Usually this imports a WSGI-compatible application which you've stored elsewhere

When you click "Next", we will create a WSGI file for you, including a simple "Hello World" app which you can use to get started, as well as some comments on how to use other frameworks.

You will also be able to specify a *virtualenv* to use for your app.

Cancel

« Back

Next »

Paso 4: Seleccionar virtualenv

- Bajar hasta la sección `virtualenv`
- Agregar el `virtualenv venv` creado anteriormente
 - `/home/usuario/.virtualenvs/venv`

xqazprog.pythonanywhere.com

+ Add a new web app

Configuration for xqazprog.pythonanywhere.com

Reload:

Reload xqazprog.pythonanywhere.com

Best before date:

We're happy to host your free website -- and keep it free -- for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Saturday 04 May 2024**

Run until 3 months from today

[Paying users'](#) sites stay up forever without any need to log in to keep them running.

Traffic:

How busy is your site?

This month (previous month)	1	(0)
Today (yesterday)	1	(0)
Hour (previous hour)	1	(0)

Want some more data? [Paying accounts](#) get pretty charts ;-)

Code:

What your site is running.

Source code: [Enter the path to your web app source code](#)


Working directory: [/home/xqazprog/](#) [Go to directory](#)

WSGI configuration file: [/var/www/xqazprog.pythonanywhere.com_wsgi.py](#)

Python version: 3.10 [🔧](#)

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here.](#) You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[Enter path to a virtualenv, if desired](#) 

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here.](#) You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.


[/home/xqazprog/.virtualenvs/venv](#)

[Start a console in this virtualenv](#) 

3. Configuración de webapp en servidor


Paso 1: Clonar repositorio privado de github

- `git clone https://[username]:[token]@github.com/[username]/[repo]`
- `git clone https://Gabo-araya:ghp_TOKEN_GENERADO_EN_GITHUB@github.com/Gabo-araya/repositorio`



pythonanywhere

by ANACONDA










/home/xqazprog/  coderberg

[Dashboard](#)
[Consoles](#)
[Files](#)
[Web](#)
[Tasks](#)
[Databases](#)

[Open Bash console here](#)
94% full – 479.7 MB of your 512.0 MB quota
[More Info](#)

























Directories


[New directory](#)

- [.git/](#) 
- [blog/](#) 
- [core/](#) 
- [login/](#) 
- [media/](#) 
- [panel/](#) 
- [static/](#) 
- [staticfiles/](#) 
- [staticfiles-cdn/](#) 

Files

[New file](#)

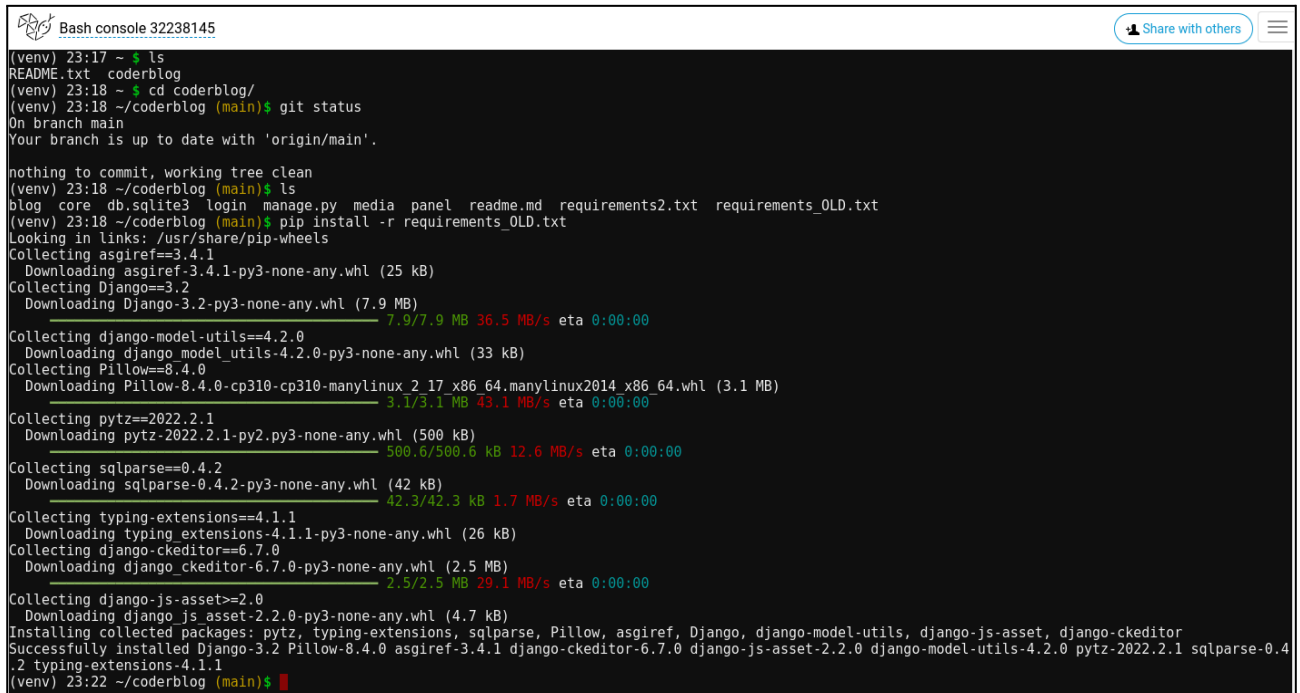
 .gitignore	   2024-02-05 01:48	2.5 KB
 db.sqlite3	   2024-02-05 01:48	216.0 KB
 manage.py	   2024-02-05 01:48	660 bytes
 readme.md	   2024-02-05 01:48	5.1 KB
 requirements.txt	   2024-02-05 01:48	167 bytes
 requirements2.txt	   2024-02-05 01:48	2.6 KB

 [Upload a file](#)

100MiB maximum size

Paso 2: Instalar los requerimientos

- `pip install -r requirements.txt`



The screenshot shows a terminal window titled "Bash console 32238145" with a "Share with others" button in the top right corner. The terminal output shows the following sequence of commands and results:

```
(venv) 23:17 ~ $ ls
README.txt  coderblog
(venv) 23:18 ~ $ cd coderblog/
(venv) 23:18 ~/coderblog (main)$ git status
On branch main
Your branch is up to date with 'origin/main'.

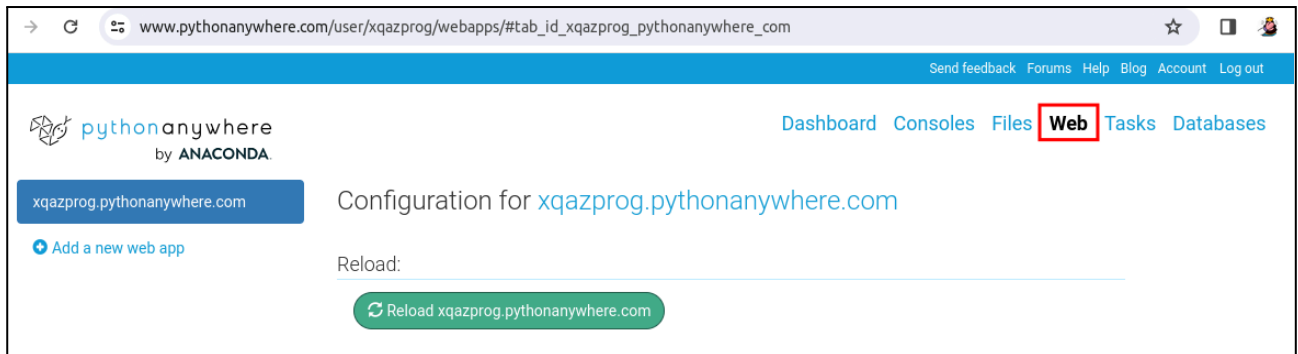
nothing to commit, working tree clean
(venv) 23:18 ~/coderblog (main)$ ls
blog  core  db.sqlite3  login  manage.py  media  panel  readme.md  requirements2.txt  requirements_OLD.txt
(venv) 23:18 ~/coderblog (main)$ pip install -r requirements_OLD.txt
Looking in links: /usr/share/pip-wheels
Collecting asgiref==3.4.1
  Downloading asgiref-3.4.1-py3-none-any.whl (25 kB)
Collecting Django==3.2
  Downloading Django-3.2-py3-none-any.whl (7.9 MB)
    7.9/7.9 MB 36.5 MB/s eta 0:00:00
Collecting django-model-utils==4.2.0
  Downloading django_model_utils-4.2.0-py3-none-any.whl (33 kB)
Collecting Pillow==8.4.0
  Downloading Pillow-8.4.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
    3.1/3.1 MB 43.1 MB/s eta 0:00:00
Collecting pytz==2022.2.1
  Downloading pytz-2022.2.1-py2.py3-none-any.whl (500 kB)
    500.6/500.6 kB 12.6 MB/s eta 0:00:00
Collecting sqlparse==0.4.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
    42.3/42.3 kB 1.7 MB/s eta 0:00:00
Collecting typing-extensions==4.1.1
  Downloading typing_extensions-4.1.1-py3-none-any.whl (26 kB)
Collecting django-ckeditor==6.7.0
  Downloading django_ckeditor-6.7.0-py3-none-any.whl (2.5 MB)
    2.5/2.5 MB 29.1 MB/s eta 0:00:00
Collecting django-js-asset>=2.0
  Downloading django_js_asset-2.2.0-py3-none-any.whl (4.7 kB)
Installing collected packages: pytz, typing-extensions, sqlparse, Pillow, asgiref, Django, django-model-utils, django-js-asset, django-ckeditor
Successfully installed Django-3.2 Pillow-8.4.0 asgiref-3.4.1 django-ckeditor-6.7.0 django-js-asset-2.2.0 django-model-utils-4.2.0 pytz-2022.2.1 sqlparse-0.4.2 typing-extensions-4.1.1
(venv) 23:22 ~/coderblog (main)$
```

4. Configurar static files

Revisar: How to setup static files in Django:

<https://help.pythonanywhere.com/pages/DjangoStaticFiles>

Paso 1: Acceder a la pestaña web



Paso 2. Bajar hasta la sección "Static files"

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
Enter URL	Enter path	

Agregar datos de static



- **URL:** /staticfiles-cdn/
- **Directory:** /home/xqazprog/coderblog/staticfiles-cdn/

Agregar datos de media

- **URL:** /media/
- **Directory:** /home/xqazprog/coderblog/media/

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much faster straight off the disk if you specify them here. You need to **Reload your web app** to activate any changes you make to the mappings below.

URL	Directory	Delete
/staticfiles-cdn/	/home/xqazprog/coderblog/staticfiles-cdn/	
/media/	/home/xqazprog/coderblog/media/	
Enter URL	Enter path	

Paso 3. Configurar rutas estáticas en settings.py

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'
STATICFILES_DIRS = [
    BASE_DIR / 'static', #os.path.join(BASE_DIR, 'static')
    os.path.join(BASE_DIR, 'static'),
]

STATIC_ROOT = BASE_DIR / 'staticfiles-cdn'
STATICFILES_STORAGE = 'whitenoise.storage.CompressedManifestStaticFilesStorage'

MEDIA_URL = 'media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media/')
```

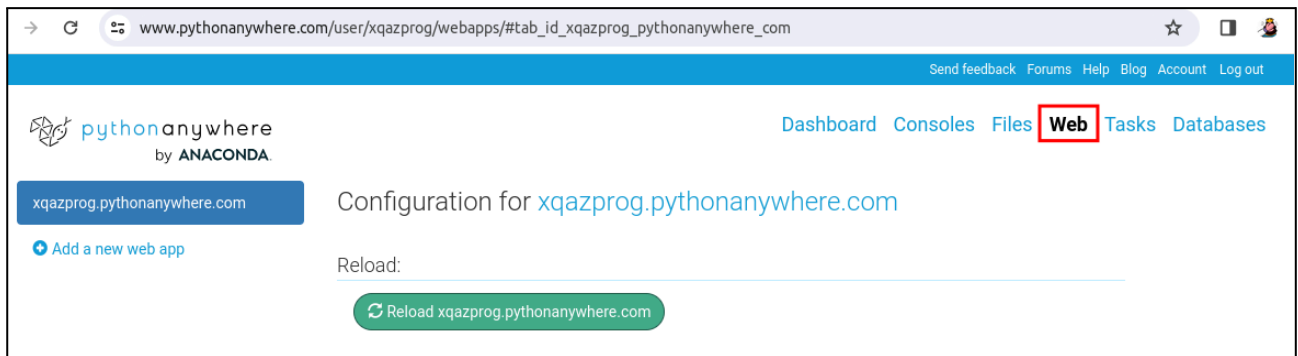
Paso 4. Recolectar estáticos

Luego, en la terminal insertar el comando para recolectar los archivos estáticos

- `python manage.py collectstatic`

5. Modificar WSGI

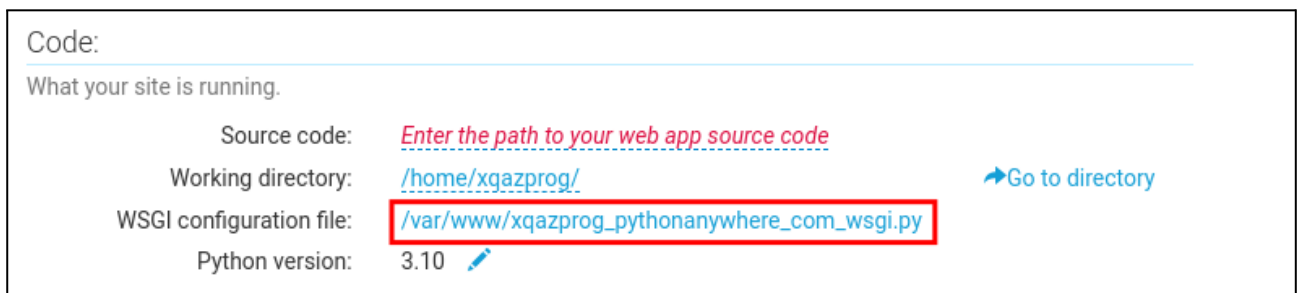
Paso 1. Acceder a la pestaña web



Paso 2. Bajar hasta la sección "Code"

Pinchar en WSGI configuration file para editar:

- `/var/www/proyecto_pythonanywhere_com_wsgi.py`



Paso 3. Editar archivo de configuración wsgi

Bajar hasta la sección de Django

- (Opcional): eliminar el código comentado no usado
- Estructura de path
 - `path = /home/[usuario]/[proyecto]/`
- Asegurarse de cambiar `DJANGO_SETTINGS_MODULE`
 - `os.environ['DJANGO_SETTINGS_MODULE'] = '[proyecto].settings'`

Descomentar las líneas comentadas hasta conseguir un código similar al siguiente:

```
# ++++++ DJANGO ++++++
# To use your own django app use code like this:
import os
import sys

## assuming your django settings file is at
'/home/usuario/proyecto/core/settings.py'
## and your manage.py is is at '/home/usuario/proyecto/manage.py'
#path = '/home/[usuario]/[proyecto]'

## assuming your django settings file is at
'/home/xqazprog/coderblog/core/settings.py'
## and your manage.py is is at '/home/xqazprog/coderblog/manage.py'
path = '/home/xqazprog/coderblog'
if path not in sys.path:
    sys.path.append(path)

#os.environ['DJANGO_SETTINGS_MODULE'] = '[project].settings'
os.environ['DJANGO_SETTINGS_MODULE'] = 'core.settings'

## then:
from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()
```

← → ↻ 🔍 www.pythonanywhere.com/user/xqazprog/files/var/www/xqazprog_pythonanywhere_com_wsgi.py?edit



/var/www/xqazprog_pythonanywhere_com_wsgi.py

```
1 # This file contains the WSGI configuration required to serve up your
2 # web application at http://xqazprog.pythonanywhere.com/
3 # It works by setting the variable 'application' to a WSGI handler of some description.
4
5 # ++++++ GENERAL DEBUGGING TIPS ++++++
6 # getting imports and sys.path right can be fiddly!
7 # We've tried to collect some general tips here:
8 # https://help.pythonanywhere.com/pages/DebuggingImportError
9
10 # ++++++ DJANGO ++++++
11 # To use your own django app use code like this:
12 import os
13 import sys
14
15 ## assuming your django settings file is at '/home/usuario/proyecto/core/settings.py'
16 ## and your manage.py is is at '/home/usuario/proyecto/manage.py'
17 #path = '/home/[usuario]/[proyecto]'
18
19 ## assuming your django settings file is at '/home/xqazprog/coderblog/core/settings.py'
20 ## and your manage.py is is at '/home/xqazprog/coderblog/manage.py'
21 path = '/home/xqazprog/coderblog'
22 if path not in sys.path:
23     sys.path.append(path)
24
25 #os.environ['DJANGO_SETTINGS_MODULE'] = '[project].settings'
26 os.environ['DJANGO_SETTINGS_MODULE'] = 'core.settings'
27
28 ## then:
29 from django.core.wsgi import get_wsgi_application
30 application = get_wsgi_application()
31
```

6. Otras configuraciones

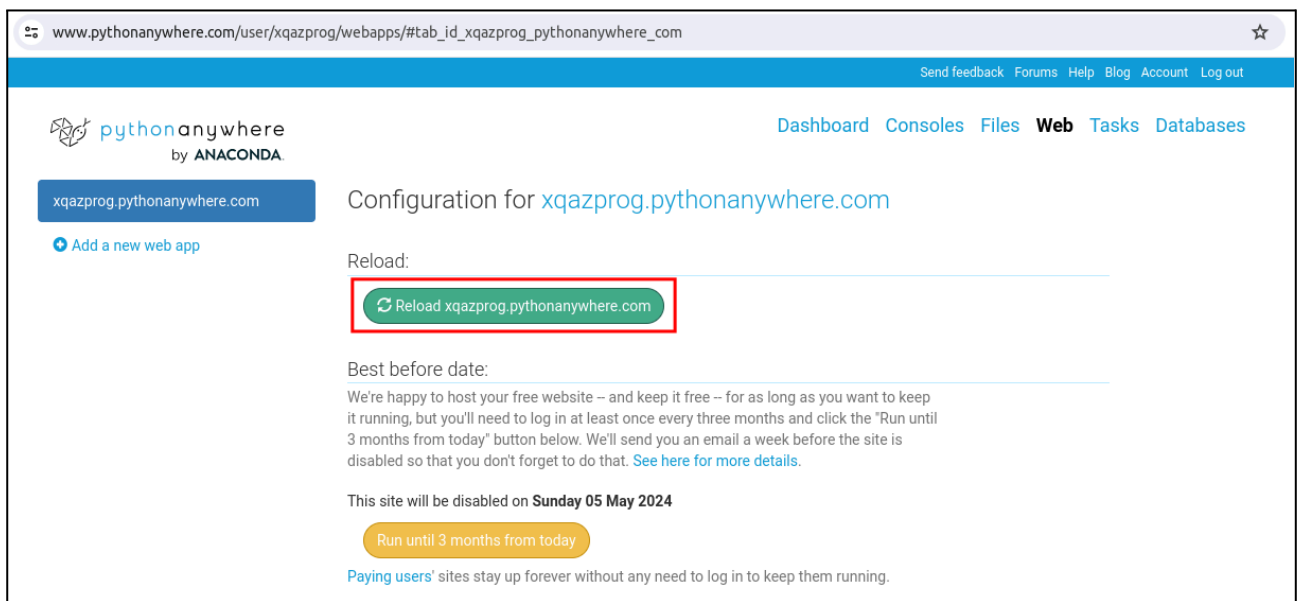
Modificar settings.py

- Agregar "proyecto.pythonanywhere.com" dentro de `allowed_hosts`
 - `ALLOWED_HOSTS = ['proyecto.pythonanywhere.com']`
- Cambiar `DEBUG = False` a `DEBUG = True`

Recargar web app

- <https://www.pythonanywhere.com/user/xqazprog/webapps/>
- Presionar botón "**Reload**" `xqazprog.pythonanywhere.com`
- Finalmente, para acceder al sitio del proyecto:
 - <https://xqazprog.pythonanywhere.com/>

Importante: Cada vez que se hacen cambios a las configuraciones es necesario recargar la webapp.



7. Otra info

- **Deploying an existing Django project on PythonAnywhere:**
<https://help.pythonanywhere.com/pages/DeployExistingDjangoProject/>
- **Using MySQL:** <https://help.pythonanywhere.com/pages/UsingMySQL/>
- **How To Migrate Sqlite To MySql Database In Django On PythonAnyWhere:**
<https://www.youtube.com/watch?v=e8fD9IU6qy4>