

# Instrucciones para el despliegue

---

<b>Nombre del Proyecto</b>	PetFun
<b>Código del Proyecto</b>	2025-3.14
<b>Fecha de Elaboración</b>	23/11/2025

Esta guía explica primero como correr el proyecto en local y luego cómo desplegar el proyecto PetFun para su puesta en producción.

## Índice

---

- A. Guía de instalación y migración (PetFun)
- B. Guía de despliegue de proyecto

## A. Guía de instalación y migración (PetFun)

---

Esta guía explica cómo preparar el entorno, aplicar migraciones y poner datos de prueba para el proyecto PetFun.

### Requisitos

- Python 3.12
- pip
- PowerShell (Windows)
- Opcional: VS Code con la extensión de Python

### 1. Clonar el repositorio

```
git clone https://github.com/Manuelgithuv/PetFun.git  
cd PetFun
```

### 2. Crear y activar entorno virtual

```
python -m venv .venv  
.venv\Scripts\Activate.ps1
```

Si PowerShell bloquea la ejecución de scripts, habilita:

```
Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned  
.\\.venv\\Scripts\\Activate.ps1
```

### 3. Instalar dependencias

```
pip install -r requirements.txt
```

### 4. Configurar variables (desarrollo)

- `DEBUG` está activado por defecto.
- Base de datos SQLite lista por defecto (`db.sqlite3`).
- Soporte `.env`: el proyecto carga variables desde un archivo `.env` en la raíz.

### 5. Migraciones (base de datos)

Si ya existe una base de datos previa y vas a cambiar de modelo de usuario, borra `db.sqlite3` antes de migrar:

```
if (Test-Path .\\db.sqlite3) { Remove-Item .\\db.sqlite3 -Force }
```

Aplica migraciones desde cero:

```
python manage.py migrate
```

Esto creará:

- Tablas del sistema (`auth`, `admin`, `sessions`, etc.).
- El modelo de usuario personalizado (`accounts.User`).
- Usuarios de prueba gracias a la migración `accounts/0002_seed_users.py`:
  - `demo1@example.com / demo12345`
  - `demo2@example.com / demo12345`

### 6. Crear superusuario (opcional)

```
python manage.py createsuperuser
```

### 7. Ejecutar el servidor de desarrollo

```
python manage.py runserver
```

- Home: <http://127.0.0.1:8000/>
- Login: <http://127.0.0.1:8000/login/>
- Registro: <http://127.0.0.1:8000/register/>
- Mi cuenta: <http://127.0.0.1:8000/account/>
- Admin: <http://127.0.0.1:8000/admin/>

## 8. Datos de prueba

- Usa los usuarios demo incluidos o crea nuevos desde [/register](#) o el panel [/admin](#).
- En [/account/](#) puedes editar el perfil, cambiar contraseña y eliminar la cuenta (con confirmación por contraseña).

### Catálogo y filtros

- Catálogo: <http://127.0.0.1:8000/catalog/>
- Puedes filtrar por categoría padre y subcategoría con query string:
  - [?parent=<NombreCategoria>](#)
  - [?parent=<NombreCategoria>&sub=<NombreSubcategoria>](#)
  - Ejemplo: [/catalog/?parent=Juguetes%20para%20Perro&sub=Mordedores](#)
- En la home verás tarjetas por categoría con accesos rápidos a subcategorías.

### Checkout y cesta

- No se puede iniciar el checkout con la cesta vacía. Si intentas ir a pagar sin artículos, serás redirigido al inicio con un aviso.

## 9. Tests y cobertura

Ejecuta los tests con cobertura:

```
python manage.py test
```

Si deseas cobertura detallada, hay que ejecutarlo manualmente:

```
coverage run manage.py test
coverage html
Abre htmlcov/index.html en tu navegador
```

## 10. Problemas comunes

- Error de migraciones inconsistentes tras cambiar `AUTH_USER_MODEL`:
  - Borra `db.sqlite3` y vuelve a `python manage.py migrate`.
- No puedo activar el entorno en PowerShell:
  - Revisa `Set-ExecutionPolicy` como arriba.
- No puedo iniciar sesión con email:

- Asegúrate de que el usuario tenga el campo `email` relleno y usa la contraseña correcta.

## 11. Uso de archivo .env

Para evitar configurar variables en cada sesión, puedes usar un archivo `.env`:

2. Copia el ejemplo y edítalo:

```
Copy-Item .env.example .env
```

3. Abre `.env` y ajusta valores (por ejemplo, claves de Stripe, SMTP, etc.).

Variables frecuentes en `.env`:

- `DEBUG=1`
- `STRIPE_SECRET_KEY=sk_test_xxx`
- `STRIPE_PUBLISHABLE_KEY=pk_test_xxx`
- `EMAIL_BACKEND=django.core.mail.backends.smtp.EmailBackend`
- `EMAIL_HOST=smtp.gmail.com / EMAIL_PORT=587 / EMAIL_USE_TLS=1`
- `EMAIL_HOST_USER=tu_correo / EMAIL_HOST_PASSWORD=tu_password_o_app_password`
- `DEFAULT_FROM_EMAIL=no-reply@petfun.local`

## B. Guía de despliegue de proyecto

---

### Requisitos

- Tener una cuenta creada en [pythonanywhere](#)

### 1. Abrir un terminal / consola de comandos

Vaya al apartado Consoles → Bash. Tras hacer este paso, se encontrara en una página parecida a esta:



Esta consola permanecerá abierta siempre y cuando no la borre, esta se encuentra de forma más accesible en el apartado Dashboard

## 2. Clonar el repositorio

Tras esto, procedemos a clonar el repositorio de git:

```
git clone https://github.com/Manuelgithuv/PetFun.git
```

y ejecutamos el comando "ls" para comprobar que se ha descargado correctamente:

```
ls
```

Debería quedar la consola tal que:



## 3. Crear un entorno virtual

Creamos un entorno virtual mediante el comando:

```
mkvirtualenv --python=/usr/bin/python3.12 .venv
```

Tras esto, en la propia terminal se verá como se ha activado el entorno creado:

```

Bash console 43660936
15:44 ~ $ git clone https://github.com/Manuelgithuv/PetFun.git
Cloning into 'PetFun'...
remote: Enumerating objects: 399, done.
remote: Counting objects: 100% (399/399), done.
remote: Compressing objects: 100% (283/283), done.
remote: Total 399 (delta 172), reused 334 (delta 107), pack-reused 0 (from 0)
Receiving objects: 100% (399/399), 2.87 MB | 4.30 MB/s, done.
Resolving deltas: 100% (172/172), done.
Updating files: 100% (164/164), done.
15:51 ~ $ ls
PetFun README.txt
15:51 ~ $ mkvirtualenv --python=/usr/bin/python3.12 .venv
created virtual environment CPython3.12.final.0-64 in 11478ms
  creator CPython3Posix(dest=/home/josemaMG/.virtualenvs/.venv, clear=False, no_vcs_ignore=False, global=False)
  seeders FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/josemaMG/.local/share/virtualenvs)
    added seed packages: pip==23.3
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/preactivate
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/postactivate
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/get_env_details
(.venv) 15:59 ~ $
```

## 4. Instalar las dependencias

Nos metemos dentro del proyecto y instalamos todas las dependencias necesarias establecidas en el archivo requirements.txt:

```

cd PetFun
pip install -r requirements.txt

```

Tras esto, en la propia terminal se verá como se han instalado todas las dependencias necesarias:

```

Bash console 43660936
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/postactivate
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/get_env_details
(.venv) 15:59 ~ $ cd PetFun
(.venv) 16:03 ~/PetFun (main)$ pip install -r requirements.txt
Collecting requests==2.28.1 (from -r requirements.txt (line 1))
  Using cached requests-2.28.1-py3-none-any.whl.metadata (4.1 kB)
Collecting python-dotenv==1.0.1 (from -r requirements.txt (line 2))
  Using cached python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Collecting stripe==10.12.0 (from -r requirements.txt (line 3))
  Using cached stripe-10.12.0-py3-none-any.whl.metadata (2.7 kB)
Collecting coverage==7.3.0 (from -r requirements.txt (line 4))
  Using cached coverage-7.3.0-cp312-cp312-manylinux_2_5_x86_64_manylinux_2_17_x86_64_manylinux2014_x86_64.whl.metadata (8.1 kB)
Collecting django_coverage_plugin==3.1.0 (from -r requirements.txt (line 5))
  Using cached django_coverage_plugin-3.1.0-py3-none-any.whl.metadata (8.9 kB)
Collecting Pillow==10.0.0 (from -r requirements.txt (line 6))
  Using cached pillow-10.0.0-cp312-cp312-manylinux_2_28_x86_64_manylinux2014_x86_64.whl.metadata (8.8 kB)
Collecting asgiref==3.8.3 (from Django==5.2.8->r requirements.txt (line 1))
  Using cached asgiref-3.11.0-py3-none-any.whl.metadata (0.3 kB)
Collecting sqlparse==0.3.1 (from Django==5.2.8->r requirements.txt (line 1))
  Using cached sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Collecting requests==2.28.1 (from stripe==10.12.0->r requirements.txt (line 3))
  Using cached requests-2.28.1-py3-none-any.whl.metadata (4.9 kB)
Collecting typing_extensions==4.15.0-py3-none-any.whl.metadata (3.3 kB)
  Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting charset_normalizer<4,>=2 (from requests>=2.20->stripe==10.12.0->r requirements.txt (line 3))
  Using cached charset_normalizer-3.4.4-cp312-cp312-manylinux2014_x86_64_manylinux_2_17_x86_64_manylinux2014_x86_64.whl.metadata (37 kB)
Collecting idna<3,>=2.5 (from requests>=2.20->stripe==10.12.0->r requirements.txt (line 3))
  Using cached idna-3.1.2-py3-none-any.whl.metadata (6.5 kB)
Collecting urllib3<2.20,>=1.26 (from requests>=2.20->stripe==10.12.0->r requirements.txt (line 3))
  Using cached urllib3-2.5.0-py3-none-any.whl.metadata (6.5 kB)
Collecting certifi==2023.4.17 (from requests>=2.20->stripe==10.12.0->r requirements.txt (line 3))
  Using cached certifi-2023.11.12-py3-none-any.whl.metadata (2.5 kB)
Using cached django-5.2.8-py3-none-any.whl (8.3 MB)
Using cached python_dotenv==1.0.1-py3-none-any.whl (1.9 kB)
Using cached stripe-10.12.0-py3-none-any.whl (1.6 MB)
Using cached coverage-7.3.0-cp312-cp312-manylinux_2_5_x86_64_manylinux_2_17_x86_64_manylinux2014_x86_64.whl (233 kB)
Using cached django_coverage_plugin-3.1.0-py3-none-any.whl (14 kB)
Using cached pillow-12.0.0-cp312-cp312-manylinux_2_27_x86_64_manylinux_2_28_x86_64.whl (7.0 MB)
Using cached asgiref-3.11.0-py3-none-any.whl (24 kB)
Using cached requests==2.28.1 (from stripe==10.12.0->r requirements.txt (line 3))
  Using cached requests-2.28.1-py3-none-any.whl (64 kB)
Using cached django_coverage_plugin-3.1.0-py3-none-any.whl (14 kB)
Using cached idna-3.1.2-py3-none-any.whl (71 kB)
Using cached urllib3-2.5.0-py3-none-any.whl (159 kB)
Using cached certifi-2023.11.12-py3-none-any.whl (44 kB)
Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Successfully installed Django==5.2.8 Pillow==12.0.0 asgiref==3.11.0 certifi==2023.11.12 charset_normalizer==3.4.4 coverage==7.3.0 django-coverage-plugin==3.1.0 idna==3.11
python-dotenv==1.0.1 requests==2.32.5 sqlparse==0.5.3 stripe==10.12.0 typing_extensions==4.15.0 urllib3==2.5.0
(.venv) 16:05 ~/PetFun (main)$
```

## 5. Configurar el "Web app" en PythonAnywhere

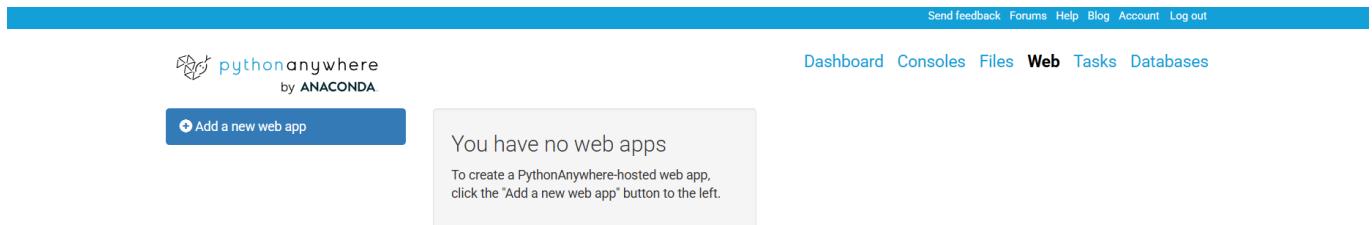
Seleccione las barras de la esquina superior derecha y vaya al apartado "Web"

```

virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/postactivate
virtualenvwrapper.user_scripts creating /home/josemaMG/.virtualenvs/.venv/bin/get_env_details
(.venv) 15:59 ~ $ cd Petfun
(.venv) 16:03 ~/Petfun (main)$ pip install -r requirements.txt
Looking in links: /usr/share/pip-wheels
Collecting requests==2.27.0 (from -r requirements.txt (line 1))
  Using cached requests-2.27.0-py3-none-any.whl.metadata (4.1 kB)
Collecting python-dotenv==1.0.1 (from -r requirements.txt (line 2))
  Using cached python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
Collecting stripe==10.12.0 (from -r requirements.txt (line 3))
  Using cached stripe-10.12.0-py3-none-any.whl.metadata (2.7 kB)
Collecting coverage==7.3.0 (from -r requirements.txt (line 4))
  Using cached coverage-7.3.0-cp312-cp312-manylinux2_5_36_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (8.1 kB)
Collecting django-coverage-plugin==3.1.0 (from -r requirements.txt (line 5))
  Using cached django_coverage_plugin-3.1.0-py3-none-any.whl.metadata (8.9 kB)
Collecting Pillow==12.0.0 (from -r requirements.txt (line 6))
  Using cached pillow-12.0.0-cp312-manylinux2_5_36_64.whl.metadata (8.8 kB)
Collecting asdf==3.1.1 (from -r requirements.txt (line 7))
  Using cached asdf-3.1.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from Django==3.2.8->-r requirements.txt (line 8))
  Using cached sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Collecting requests>=2.20 (from stripe==10.12.0->-r requirements.txt (line 9))
  Using cached requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting typing_extensions==3.10.0 (from stripe==10.12.0->-r requirements.txt (line 10))
  Using cached typing_extensions-3.10.0-py3-none-any.whl.metadata (2.3 kB)

```

Ahora en esta pantalla clickeé en el apartado "Add a new web app"



En esta pantalla se abrirá un panel, pulse el botón "Next". Tras esto, el panel cambiará y ahora tendrá que seleccionar la opción "Manual configuration". Despues clickeé en "Python 3.12" y tras esto en "Next".

Tras esto se creara la web "[tu usuario].pythonanywhere.com".

En la sección de Virtualenv de esa web app, indique la ruta de su virtualenv (/home/tu usuario/.virtualenvs/.venv) y guárdalo.

### Traffic:

How busy is your site?

This month (previous month)	911	(0)
Today (yesterday)	55	(26)
Hour (previous hour)	1	(18)

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/josemaMG/](#)

[Go to directory](#)

WSGI configuration file: [/var/www/josemamg\\_pythonanywhere\\_com\\_wsgi.py](#)

Python version: 3.12

### 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](#)

Debería de quedar algo tal que así:

---

Traffic:

How busy is your site?

This month (previous month)	911	(0)
Today (yesterday)	55	(26)
Hour (previous hour)	1	(18)

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

---

Code:

What your site is running.

Source code:	<a href="#"><i>Enter the path to your web app source code</i></a>
Working directory:	<a href="/home/josemaMG/">/home/josemaMG/</a>
WSGI configuration file:	<a href="/var/www/josemamg_pythonanywhere_com_wsgi.py">/var/www/josemamg_pythonanywhere_com_wsgi.py</a>
Python version:	3.12 

 [Go to directory](#)

---

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/josemaMG/.virtualenvs/.venv>

 [Start a console in this virtualenv](#)

## 6. Editar el archivo WSGI para apuntar a tu proyecto

Desde la propia pestaña Web, haga clic en el enlace del archivo WSGI configuration file (algo como /var/www/tuusuario\_pythonanywhere\_com\_wsgi.py). Abra el editor.

## Traffic:

How busy is your site?

This month (previous month)	911	(0)
Today (yesterday)	55	(26)
Hour (previous hour)	1	(18)

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/josemaMG/](#) [↗ Go to directory](#)  
WSGI configuration file: [/var/www/josemamg\\_pythonanywhere\\_com\\_wsgi.py](#)  
Python version: 3.12 [Edit](#)

## 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/josemaMG/.virtualenvs/venv](#)

[🔗 Start a console in this virtualenv](#)

Borre lo que haya en el archivo y ponga. **OJO, donde ponga "tuusuario" ponga su nombre de usuario, si no, no funcionara:**

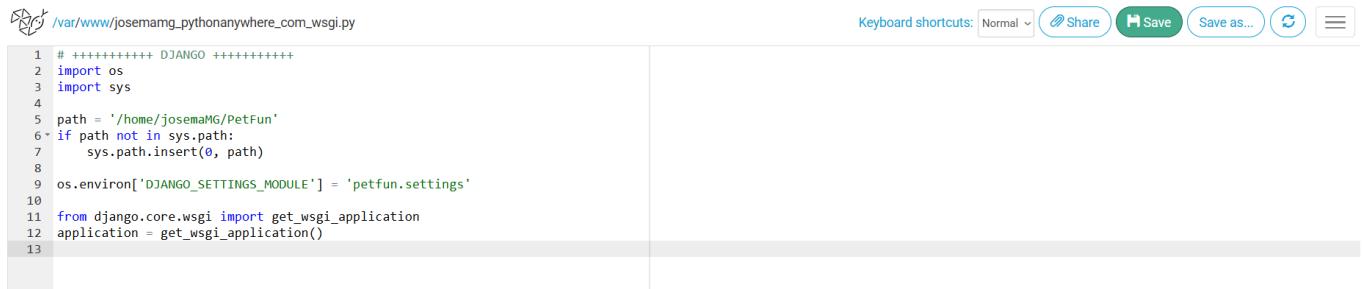
```
# ++++++ DJANGO ++++++
import os
import sys

path = '/home/tuusuario/PetFun'
if path not in sys.path:
    sys.path.insert(0, path)

os.environ['DJANGO_SETTINGS_MODULE'] = 'petfun.settings'

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

Dentro del archivo modificado con mi nombre de usuario queda un archivo tal que:



```

1 # ++++++ DJANGO ++++++
2 import os
3 import sys
4
5 path = '/home/josemamg/PetFun'
6 if path not in sys.path:
7     sys.path.insert(0, path)
8
9 os.environ['DJANGO_SETTINGS_MODULE'] = 'petfun.settings'
10
11 from django.core.wsgi import get_wsgi_application
12 application = get_wsgi_application()
13

```

Como puede ver en la imagen ahora pulse en el botón de guardar "Save".

## 7. Establecer los archivos estáticos del proyecto.

Vuelva a darle a las tres barras horizontales (en la esquina superior derecha) y vaya a "Files". Entre en PetFun/ → petfun/ → settings.py.

En la línea 52 encontrara "ALLOWED\_HOSTS" cambie esa línea completa por esta. **OJO, donde ponga "tuusuari" ponga su nombre de usuario, si no, no funcionara:**

```
ALLOWED_HOSTS = ['tuusuari.pythonanywhere.com']
```

Justo debajo aparece las variables:

```
MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'
```

Pues copie este código ponga justo abajo. **OJO, donde ponga "tuusuari" ponga su nombre de usuario, si no, no funcionara:**

```
STATIC_URL = '/static/'

STATIC_ROOT = '/home/tuusuari/PetFun/staticfiles'

STATICFILES_DIRS = [
    '/home/tuusuari/PetFun/media/products',
]
```

El archivo tiene queda, en mi caso, tal que:

```

51 ALLOWED_HOSTS = ['josemamg.pythonanywhere.com']
52
53 MEDIA_URL = '/media/'
54 MEDIA_ROOT = BASE_DIR / 'media'
55
56 STATIC_URL = '/static/'
57
58 STATIC_ROOT = '/home/josemaMG/PetFun/staticfiles'
59
60
61 STATICFILES_DIRS = [
62     '/home/josemaMG/PetFun/media/products',
63 ]
64
65
66 # Application definition
67
68 INSTALLED_APPS = [
69     "django.contrib.admin",
70     "django.contrib.auth",
71     "django.contrib.contenttypes",
72     "django.contrib.sessions",

```

Tras esto, guarde los cambios (boton "Save") y en las tres barras horizontales, vaya a "Web" y vaya al apartado de "Virtualenv" para abrir la terminal con el entorno virtual activado:

### Traffic:

How busy is your site?

This month (previous month)	911	(0)
Today (yesterday)	55	(26)
Hour (previous hour)	1	(18)

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

### Code:

What your site is running.

Source code:	<a href="#">Enter the path to your web app source code</a>	
Working directory:	<a href="#">/home/josemaMG/</a>	<a href="#">↗ Go to directory</a>
WSGI configuration file:	<a href="#">/var/www/josemamg_pythonanywhere_com_wsgi.py</a>	
Python version:	3.12 <a href="#"></a>	

### 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/josemaMG/.virtualenvs/venv](#)

[Start a console in this virtualenv](#)

Deberia de salir esta terminal:



Entonces copie en la terminal. Esto lo que hará es crear la carpeta staticfiles **OJO, donde ponga "tuusuario" ponga su nombre de usuario, si no, no funcionara:**

```
mkdir /home/tuusuario/PetFun/staticfiles
```

En mi caso:

```
(.venv) 18:04 ~ $ mkdir /home/josemaMG/PetFun/staticfiles  
(.venv) 18:11 ~ $
```

Y por último, ejecute este comando para copiar los archivos de una carpeta a otra.

```
workon venv  
cd ~/PetFun  
python manage.py collectstatic
```

Debería quedar algo tal que:

```
Bash console in virtualenv.venv  
(.venv) 18:04 ~ $ mkdir /home/josemaMG/PetFun/staticfiles  
(.venv) 18:11 ~ $ workon .venv  
cd ~/PetFun  
python manage.py collectstatic  
149 static files copied to '/home/josemaMG/PetFun/staticfiles'.  
(.venv) 20:37 ~/PetFun (main)$
```

## 8. Realizar las migraciones.

Para realizar las migraciones solo tiene que usar el comando:

```
python manage.py migrate
```

El terminal debería de quedar como se muestra a continuación:

```
Bash console in virtualenv.venv
(.venv) 18:04 ~ $ mkdir /home/josemaMG/PetFun/staticfiles
(.venv) 18:11 ~ $ workon .venv
cd ~/PetFun
python manage.py collectstatic
149 static files copied to '/home/josemaMG/PetFun/staticfiles'.
(.venv) 20:37 ~/PetFun (main)$ python manage.py migrate
Operations to perform:
  Apply all migrations: accounts, admin, auth, cart, catalog, contenttypes, core, orders, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0001_initial... OK
  Applying auth.0002.Alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validator_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying accounts.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying catalog.0001_initial... OK
  Applying cart.0001_initial... OK
  Applying core.0001_initial... OK
  Applying orders.0001_initial... OK
  Applying orders.0002_alter_order_total... OK
  Applying sessions.0001_initial... OK
(.venv) 20:50 ~/PetFun (main)$
```

## 9. Establecer el .env.

Vuelva a darle a las tres barras horizontales (en la esquina superior derecha) y vaya a "Files" para acto seguido ir a PetFun/.

Dentro tiene que crear el archivo vacío ".env"

The screenshot shows a file manager interface with two tabs: "Directories" and "Files". In the "Files" tab, there is a single file named ".env". The "Directories" tab shows several subfolders: .git/, Images/, accounts/, cart/, catalog/, core/, docs/, media/, orders/, petfun/, staticfiles/, and templates/. There is also a "New directory" button. At the bottom, there is a "Upload a file" button and a note that says "100MB maximum size".

Metase dentro del archivo y pegue lo siguiente:

```
DEBUG=1
ALLOWED_HOSTS='josemamg.pythonanywhere.com'
SECRET_KEY=change-me

STRIPE_SECRET_KEY=
STRIPE_PUBLISHABLE_KEY=pk_test_51SSkv3JfoY0bVIIdPwIeMBGf1hjE8L01C05L2LL0tKhz6BFywy
nE9B1h0kkkjgS7zmnBckDcPZ52TibBEVvfhMaY00JrZyb6Ed

EMAIL_BACKEND=django.core.mail.backends.smtp.EmailBackend
EMAIL_HOST=smtp.gmail.com
```

```
EMAIL_PORT=587
EMAIL_USE_TLS=1
EMAIL_USE_SSL=0
EMAIL_HOST_USER=noreply.petfun@gmail.com
EMAIL_HOST_PASSWORD=fdat lszp oapy nnt1
DEFAULT_FROM_EMAIL=noreply.petfun@gmail.com
```

Debería de quedar tal que:

The screenshot shows a file management interface with a sidebar for 'Directories' and a main area for 'Files'. The 'Directories' sidebar lists several subfolders. The 'Files' area contains several files with their last modified dates, sizes, and icons. There are buttons for 'Open Bash console here', 'More Info', 'New file', 'Upload a file', and a note about the 100MB maximum size.

Después guardelo.

## 10. Establecer la ruta de archivos a usar y listo.

Vaya a la sección "Web" mediante el icono de las líneas horizontales y busque la sección "Static files"

### 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/josemaMG/.virtualenvs/.venv>

[Start a console in this virtualenv](#)

### Log files:

The first place to look if something goes wrong.

Access log: <josemamg.pythonanywhere.com.access.log>

Error log: <josemamg.pythonanywhere.com.error.log>

Server log: <josemamg.pythonanywhere.com.server.log>

Log files are periodically rotated. You can find old logs here: </var/log>

### 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
<a href="#">Enter URL</a>	<a href="#">Enter path</a>	

En esta tiene que poner de URL = /static/ y Directory = /home/tuusuario/PetFun/staticfiles. **OJO, donde ponga "tuusuario" ponga su nombre de usuario, si no, no funcionara**

En mi caso sería:

#### 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/josemaMG/.virtualenvs/.venv>

 [Start a console in this virtualenv](#)

#### Log files:

---

The first place to look if something goes wrong.

Access log: <josemamg.pythonanywhere.com.access.log>

Error log: <josemamg.pythonanywhere.com.error.log>

Server log: <josemamg.pythonanywhere.com.server.log>

Log files are periodically rotated. You can find old logs here: </var/log>

#### 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
<a href="/static/">/static/</a>	<a href="/home/josemaMG/PetFun/staticfiles">/home/josemaMG/PetFun/staticfiles</a>	
<a href="#">Enter URL</a>	<a href="#">Enter path</a>	

Y por último, recargar el dominio (boton "reload").

Send feedback Forums Help Blog Account Log out

 pythonanywhere  
by ANACONDA

Dashboard Consoles Files Web Tasks Databases

josemaMG.pythonanywhere.com Configuration for josemaMG.pythonanywhere.com

+ Add a new web app

Reload: [Reload josemaMG.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 **Friday 27 February 2026**

[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)	911	(0)
Today (yesterday)	55	(26)
Hour (previous hour)	0	(0)

Y listo, ¡ya estaría PetFun funcionando en su dominio personal!

 PetFun Iniciar sesión Crear cuenta Seguimiento

**Todo para tus mascotas en un solo lugar** 

[Ver Catálogo](#)

**Tu Cesta**  
Tu cesta está vacía.

Total: 0 € [Ir a pagar](#)

**Productos destacados**



Túnel con ventana  
16.50 € [Añadir](#)



Spray catnip  
5.20 € [Añadir](#)



Caña telescópica  
7.99 € [Añadir](#)