

MANUAL DE FUNCIONAMIENTO DE PYTHONANYWHERE

PythonAnywhere es un entorno de desarrollo integrado y alojamiento en la nube, diseñado especialmente para desarrolladores que trabajan con Python. Permite ejecutar scripts de Python, crear aplicaciones web con frameworks como Flask, Django, web2py y Bottle, además de gestionar tareas programadas, entre otras funcionalidades. Este manual describe cómo utilizar PythonAnywhere para desarrollar, desplegar y gestionar tus aplicaciones web en la nube. Para este proyecto usamos la opción paga de PythonAnywhere, debido a que nuestro proyecto hace uso de la API de ChatGPT, la cual requiere conexión constante a internet para su correcto funcionamiento enviando solicitudes y recibiendo respuestas desde los servidores de OpenAI. Una alternativa para usar PythonAnywhere es contando con mi propio servidor con acceso completo a internet.

A continuación, se encuentra el dashboard (panel principal), que es la pagina principal al iniciar sesión

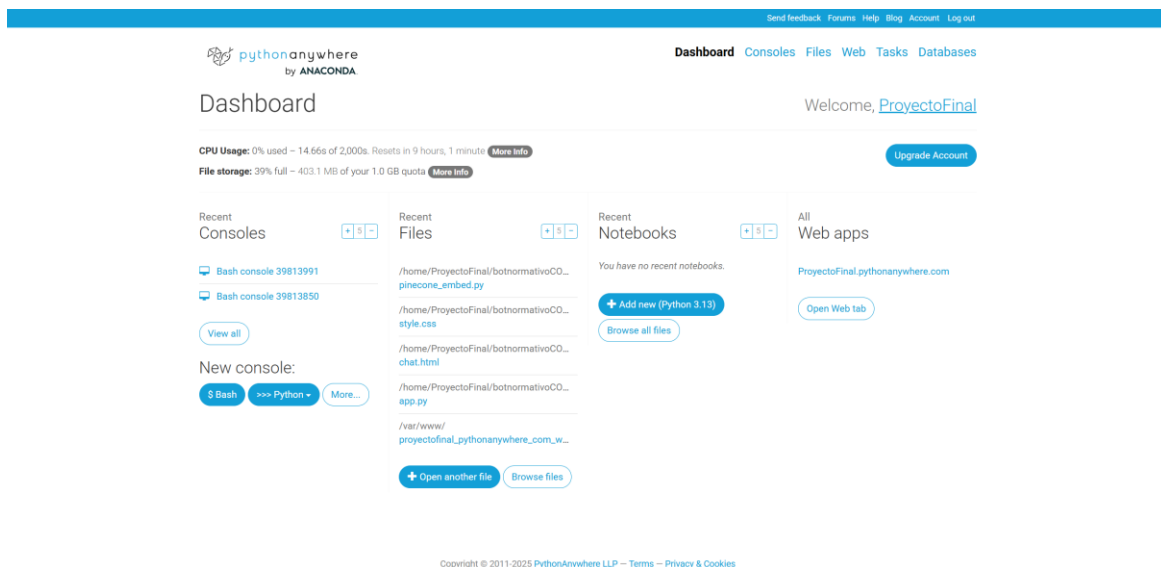


Figura 1

En la figura 2 se encuentran resaltadas las funciones del panel principal.

1. Menú principal

Permite moverse entre las secciones de la cuenta para gestionar el entorno, archivos, consola, web, tareas programadas o bases de datos.:

- Dashboard: Página principal que se ve en la figura 1
- Consoles: Se puede abrir y gestionar las consolas ya sea Bash o Python
- Files: Ver, editar o subir los archivos del proyecto
- Web: Realizar y configurar la aplicación web
- Tasks: Automatizar tareas programadas
- Databases: Gestionar las bases de datos

2. Recursos de la cuenta

- CPU Usage: Tiempo de CPU consumido. En este ejemplo con el uso del proyecto se ha usado 14.66 segundos de los 2000 segundos disponibles por día (para cuentas gratuitas).
- File Storage: Indica el espacio de almacenamiento usado. En este caso se ha utilizado 403.1 MB de 1 GB.

3. Recent Consoles

Se observa el historial de las consolas usadas, ya sean Bash o Python, además al ingresar se podrá crear una nueva consola. Aquí se instalan librerías, paquetes y se corren los scripts manualmente. En este ejemplo se puede apreciar que hay dos consolas Bash abiertas.

- Si se hace clic en el nombre de una consola (como "Bash console 39813991"), se abrirá esa terminal específica para seguir trabajando.
- Clic en **"View all"**, se encuentra una lista de todas las consolas activas y pasadas.
- Se puede crear una nueva consola desde los botones de abajo (Bash o Python).

4. Recent Files

Lista los archivos que se abrieron o modificaron. En este apartado se suben los archivos que se desean cargar a la nube, además, después de subirlos también se pueden modificar.

- Clic en un archivo (como app.py), se abre en el editor web de PythonAnywhere.
- En **"Open another file"**, se puede buscar y abrir otro archivo.
- **"Browse files"**, lleva al navegador de archivos completo.

5. Recent Notebooks

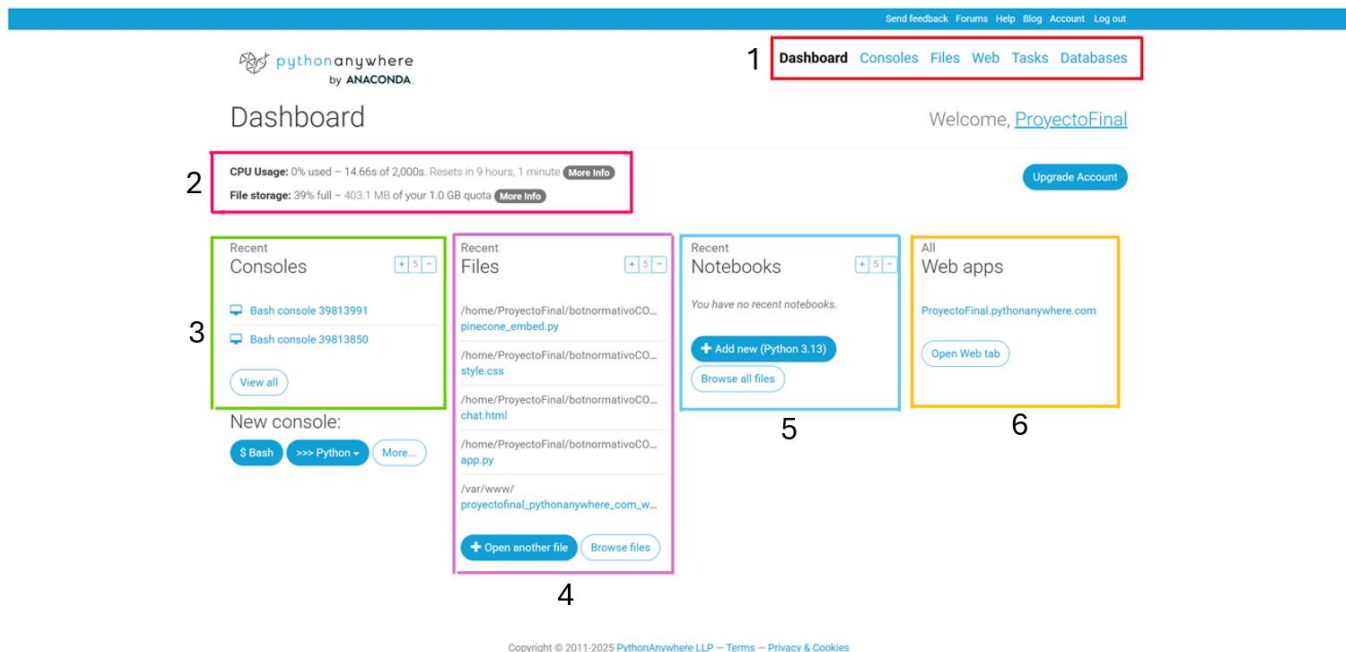
Se muestran los últimos Jupyter Notebooks usados, si se tiene alguno, estos tienen la extensión .ipynb. Un Jupyter Notebook es un entorno interactivo que permite: Escribir y ejecutar código Python por bloques, incluir texto explicativo con formato (usando Markdown), insertar gráficos, imágenes y resultados justo debajo del código, y probar funciones paso a paso, ideal para **pruebas, análisis de datos, visualizaciones o aprendizaje de Python**.

- “**Add new**”, crea un nuevo Jupyter Notebook.
- “**Browse all files**”, muestra todos los notebooks almacenados en la cuenta.

6. Web apps

Aca se listan las apps web que se tienen desplegadas y en línea.

- Clic en el **nombre del dominio** (como ProyectoFinal.pythonanywhere.com), se abre el **panel de configuración web** donde se pueden cargar archivos, reiniciar el servidor, ver logs de errores, etc.
- “**Open Web tab**”, abre la página web en una nueva pestaña del navegador para verla como usuario final.



PROCESO PARA SUBIR EL PROYECTO A PYTHONANYWHERE

REQUISITOS PREVIOS

Antes de comenzar, es importante tener:

- Una cuenta paga en PythonAnywhere (para el caso en que se necesite estar conectado a internet de manera constante)
- Todo el proyecto en una carpeta (por ejemplo, archivos .py, .html, .css, etc.).
- Saber si el proyecto es un script, una aplicación web o un Jupyter Notebook.

1. Inicio de sesión y acceso al Dashboard

El usuario debe ingresar a <https://www.pythonanywhere.com/> para iniciar sesión. Una vez dentro, se encontrará el *Dashboard* (como se muestra en la imagen). Desde este panel, se puede acceder a secciones como *Consoles*, *Files*, *Web* y más.

2. Subida de archivos del proyecto

A continuación, se hace clic en la pestaña "*Files*". En esa vista, el usuario debe ingresar al directorio `/home/tu_usuario/` o crea una nueva carpeta como `/home/tu_usuario/ProyectoFinal/`. En la parte inferior de la página se encuentra un botón llamado "*Upload a file*" mediante el cual se deben subir todos los archivos del proyecto (por ejemplo: `app.py`, `chat.html`, `style.css`, etc.), También existe la opción de subir un archivo comprimido (.ZIP) y, una vez cargado, seleccionar "*Open Bash console here*" en la parte superior, donde deberá ejecutar el siguiente comando:

```
unzip /home/tu_usuario/ruta/al/archivo.zip -d /home/tu_usuario/ruta/de/extracción
```

3. Instalación de librerías necesarias

Una vez cargados los archivos del proyecto, el usuario debe instalar las librerías necesarias para su correcto funcionamiento. Para ello, desde la pestaña *Files*, debe hacer clic en el botón *Open Bash console here*, ubicado en la parte superior de la vista del directorio.

Al abrirse la consola, se deben ejecutar los comandos de instalación correspondientes. Por ejemplo, si se cuenta con un archivo `requirements.txt`, el usuario puede instalar todas las dependencias con el siguiente comando:

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

En caso de que no exista dicho archivo, se pueden instalar las librerías individualmente. Por ejemplo:

```
pip install --user flask  
pip install --user openai
```

4. Ejecución de un archivo desde la consola

Después de instalar las librerías necesarias, el usuario puede ejecutar su archivo principal desde la consola. Para ello, debe asegurarse de estar ubicado en el directorio correcto donde se encuentra el archivo (por ejemplo, `app.py`).

Desde la pestaña Files, se hace clic en *Open Bash console here*. Una vez abierta la consola y estando en el directorio adecuado, se utiliza el siguiente comando para ejecutar el archivo:

```
python3 app.py
```

Nota: Si algo no funciona correctamente al ejecutar el archivo, el usuario puede editar el contenido directamente desde PythonAnywhere. Para hacerlo, debe ir a la pestaña *Files*, buscar el archivo en cuestión y hacer clic en el botón *Edit* para modificar el código en línea.

5. Configurar la aplicación web en PythonAnywhere

Después de subir los archivos y asegurarse de que las librerías estén instaladas, el usuario debe ir a la pestaña Web en el Dashboard y hacer clic en *Add a new web app*. En el asistente de configuración:

- Elegir la opción Manual configuration.
- Seleccionar la versión de Python que corresponde al proyecto (por ejemplo, Python 3.10).
- En el campo Source code, escribir la ruta de la carpeta del proyecto (por ejemplo: `/home/tu_usuario/MiWeb/`).
- En WSGI configuration file, hacer clic en el enlace para abrir el archivo WSGI y editarlo.

Dentro del archivo WSGI, se debe reemplazar el contenido de ejemplo por una línea como esta:

```
from MiWeb.app import app as application
```

Esto supone que el archivo principal se llama `app.py` y que está dentro de la carpeta `MiWeb`.

6. Probar y editar la página web

Después de guardar los cambios en el archivo WSGI, se debe hacer clic en el botón *Reload* dentro de la pestaña *Web* para reiniciar la aplicación. Una vez recargada, se puede abrir la URL pública que proporciona PythonAnywhere para ver la página web en funcionamiento.

Nota: Si se desea cambiar algo en el diseño o funcionalidad, se pueden editar los archivos directamente desde la pestaña *Files*, haciendo clic en *Edit* junto al archivo que se quiera modificar. Luego, se debe volver a recargar la aplicación desde la pestaña *Web* para que los cambios se reflejen.

7. Automatización de código en PythonAnywhere

PythonAnywhere permite programar tareas automáticas a través de la pestaña *Tasks*, lo cual es muy útil si el proyecto necesita ejecutarse sin intervención manual. Antes de automatizar cualquier script, es importante asegurarse de que el archivo se ejecute sin errores desde la consola. Para esto, se puede abrir una consola con *Open Bash console here* en la carpeta del proyecto y ejecutar el comando:

```
python3 nombre_del_archivo.py
```

8. Ir a la pestaña "Tasks"

Desde el Dashboard, el usuario debe hacer clic en la pestaña *Tasks* (en la parte superior). Esta sección permite programar scripts para que se ejecuten automáticamente en intervalos definidos. A continuación, se debe hacer clic en el botón *"Add a new scheduled task"*. Ahí aparecerán los siguientes campos que deben completarse:

- **Time:** seleccionar la hora exacta en la que se desea ejecutar el script (se puede configurar varias veces al día, o una sola vez).
- **Command to run:** aquí se escribe el comando para ejecutar el script, por ejemplo:

```
python3 /home/tu_usuario/MiWeb/script_automatico.py
```