



DESPLEGAR SITIO EN PYTHONANYWHERE

¿Qué es PythonAnywhere?

PythonAnywhere es una plataforma en la nube diseñada específicamente para ejecutar y desplegar aplicaciones escritas en Python. Permite a los desarrolladores trabajar directamente desde un navegador web sin necesidad de instalar Python o servidores en su computadora.

Crear archivo .env

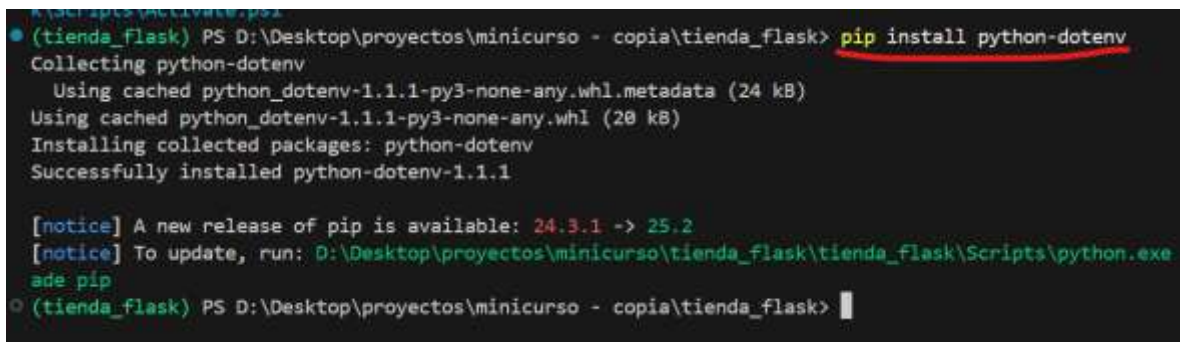
Para no subir información confidencial como credenciales de las bases de datos.



```
.env
1 #api_key
2 FLASK_SECRET_KEY= PzvYyiSmNgVlSxpOus8vnXnUzgQHfsjfaYWtJI
3
4 #credenciales para coneccion a la bd
5 DB_HOST=localhost
6 DB_NAME=prueba_flask
7 DB_USER=root
8 DB_PASSWORD=root
```

Configurar las variables de entorno

Instala la librería para manejar variables de entorno.



```
(tienda_flask) PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> pip install python-dotenv
Collecting python-dotenv
  Using cached python_dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
  Using cached python_dotenv-1.1.1-py3-none-any.whl (20 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.1.1

[notice] A new release of pip is available: 24.3.1 -> 25.2
[notice] To update, run: D:\Desktop\proyectos\minicurso\tienda_flask\tienda_flask\Scripts\python.exe
ade pip
(tienda_flask) PS D:\Desktop\proyectos\minicurso - copia\tienda_flask>
```

Modifica el archivo de conexión a la base de datos, importando la librería dontenv y os y reemplazando los valores actuales por las variables del archivo .env



```
.env × coneccion_bd.py ● app.py
coneccion_bd.py > obtener_conexion
1  import mysql.connector
2  from mysql.connector import Error
3  from dotenv import load_dotenv
4  import os
5  def obtener_conexion():
6      try:
7          conexion= mysql.connector.connect(
8              host=os.getenv('DB_HOST'),
9              database=os.getenv('DB_NAME'),
10             user=os.getenv('DB_USER'),
11             password=os.getenv('DB_PASSWORD'),
12             )
13         if conexion.is_connected():
14             print("conexion exitosa a la base de datos")
15             return conexion
16         else:
17             print("No se pudo establecer una conexion")
18             return None
19
20     except Error as e:
21         print(f"Error al conectar a la base de datos {e}")
22         return None
23
24
```

Igualmente, en el archivo principal app.py importamos las librerías anteriores y cargamos las variables de entorno.



```
app.py > ...
1  from flask import Flask, render_template, request, redirect, url_for
2  from consultas import insertar, consulta
3  from dotenv import load_dotenv
4  import os
5
6  load_dotenv()
7
8  app = Flask(__name__)
9
10 app.secret_key = os.getenv("FLASK_SECRET_KEY")
11
12 @app.route('/')
13 def inicio():
14     mensaje = "estamos aprendiendo python con flask"
15     return render_template('index.html', mensaje= mensaje)
```

Crear archivo .gitignore

Añadimos los nombres de carpeta o archivos con información sensible o innecesaria, la cual va a ser ignorada por git por lo tanto no se va a subir.

```
.gitignore
1  # archivo con las variables de entorno
2  .env
3  #script compilados de python
4  __pycache__/
5  *.pyc
6
7  #entorno virtual
8  tienda_flask/
9
10 #otros archivos
```

Generar requirements.txt

Crear archivo con las dependencias necesarias para el funcionamiento de nuestro proyecto.

Escribe en la consola `pip freeze > requirements.txt`



```
requirements.txt
1  blinker==1.9.0
2  click==8.2.1
3  colorama==0.4.6
4  flask==3.1.2
5  itsdangerous==2.2.0
6  Jinja2==3.1.6
7  MarkupSafe==2.0.2
8  mysql-connector-python==8.4.0
9  python-dotenv==1.1.1
10
11 Werkzeug==3.1.3

PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> pip freeze > requirements.txt
```

Subir proyecto a github

```
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> pip freeze > requirements.txt
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> git init
Initialized empty Git repository in D:\Desktop\proyectos\minicurso - copia\tienda_flask\.git\
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> git add .
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> git commit -m "primer commit"
[master (root-commit) 84da58] primer commit
 8 files changed, 156 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 app.py
 create mode 100644 conexion_bd.py
 create mode 100644 consultas.py
 create mode 100644 requirements.txt
 create mode 100644 templates/contacto.html
 create mode 100644 templates/index.html
 create mode 100644 templates/mensajes.html
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> git branch -m Main
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> git remote add origin https://github.com/IsrAcevedo/minicurso-flask.git
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask> git push -u origin Main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 2.49 KiB | 1.25 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/IsrAcevedo/minicurso-flask.git
 * [new branch]      Main -> Main
branch 'Main' set up to track 'origin/Main'.
PS D:\Desktop\proyectos\minicurso - copia\tienda_flask>
```

Configuración en PythonAnywhere

Crear cuenta e iniciar sesión



[Send feedback](#) [Forums](#) [Help](#) [Blog](#) [Pricing & sign up](#) [Login](#)



Host, run, and code Python in the cloud!

Get started for free. Our basic plan gives you access to machines with a full Python environment already installed. You can develop and host your website or any other code directly from your browser without having to install software or manage your own server.

Need more power? Upgraded plans start at \$5/month.

[Start running Python online in less than a minute! »](#)

[Watch our short video »](#)



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.



Create your account

Username:

Email: Did you mean iacevedo@sena.edu.co?

Password:

Password (again):

☒ I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old.

[Register](#)

We promise not to spam or pass your details on to anyone else.

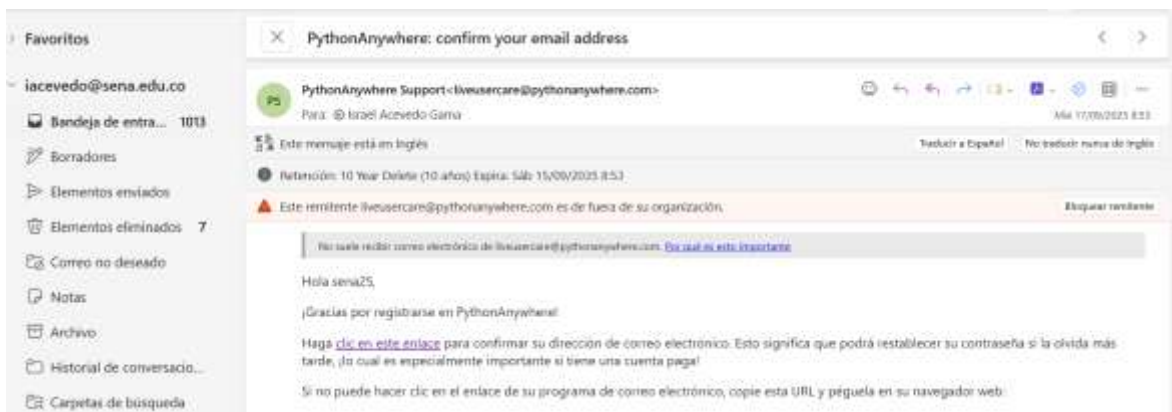


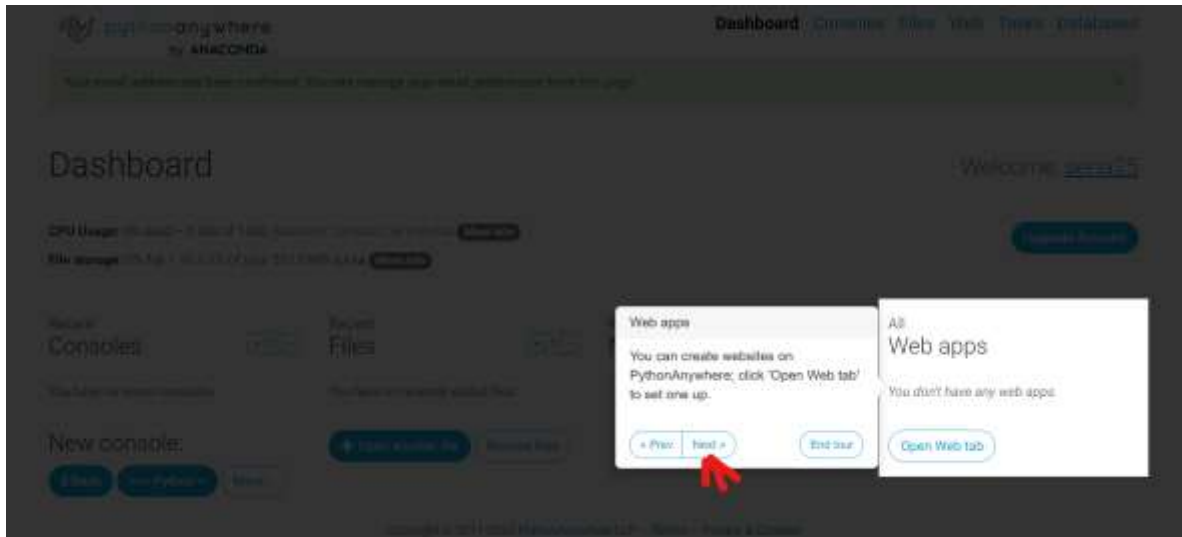
Almost there!

A confirmation link was sent to iacevedo@sena.edu.co.

All users need to have a confirmed email address to use PythonAnywhere. Check your inbox for an email from us and click the link to continue.

If this is the wrong email address or the confirmation email hasn't arrived after 5 minutes, click [here](#).







Crea una nueva aplicación web

El nombre de dominio de tu aplicación web

Su cuenta no admite nombres de dominio personalizados, por lo que su aplicación web PythonAnywhere estará disponible en `sena25.pythonanywhere.com`.

¿Quieres cambiar eso? [¡Actualiza ahora!](#)

De lo contrario, simplemente haga clic en "Siguiente" para continuar.

Cancelar

« Atrás

Siguiente »

Crea una nueva aplicación web

Seleccione un marco web de Python

...o seleccione "Configuración manual" si desea un control detallado.

- » Django
- » web2py
- » Matraz
- » Botella
- » Configuración manual (incluidos los entornos virtuales)

¿Qué otros marcos deberíamos tener aquí? ¡Envíanos tus comentarios usando el enlace en la parte superior de la página!

Cancelar

« Atrás

Siguiente »



Tengan en cuenta si tienen el traductor en español nos puede cambiar el nombre algunos nombres, en este caso nos traduce el nombre de Flask a Matraz.

Crea una nueva aplicación web

Seleccione una versión de Python

- » Python 3.9 (Flask 3.0.3)
- » Python 3.10 (Flask 3.0.3)
- » Python 3.11 (Flask 3.0.3)
- » Python 3.12 (Flask 3.0.3)
- » Python 3.13 (Flask 3.0.3)

Nota: Si desea utilizar una versión diferente de Flask a la versión predeterminada, puede hacerlo Utilice un entorno virtual para su aplicación web. Hay [instrucciones aquí](#).

Cancelar

« Atrás

Siguiente »

Seleccionamos la última versión de Python.

Crea una nueva aplicación web

Inicio rápido de un nuevo proyecto Flask

Ingresa una ruta para un archivo Python que desea utilizar para almacenar su aplicación Flask. **Si este archivo ya existe, su contenido se sobrescribirá con la nueva aplicación.**

Camino

/home/sena25/tienda/app.py

Cancelar

« Atrás

Siguiente »



Esto nos creara una carpeta llamada tienda donde se guardará el proyecto y crea el archivo principal.

Sin embargo, esto nos va hacer una configuración adicional ya que al clonar el repositorio se nos va a crear una carpeta diferente.

Otra manera de hacerlo seria primero clonar el repositorio para que se nos cree dicha carpeta y luego si podemos apuntar hacia esta carpeta.

pythonanywhere
by ANACONDA

Dashboard Consoles Files Web Tasks Databases

All done! Your web app is now set up. Details below.

sena25.pythonanywhere.com

Add a new web app.

Configuration for sena25.pythonanywhere.com

Reload:

Reload sena25.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 19 December 2025**

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?

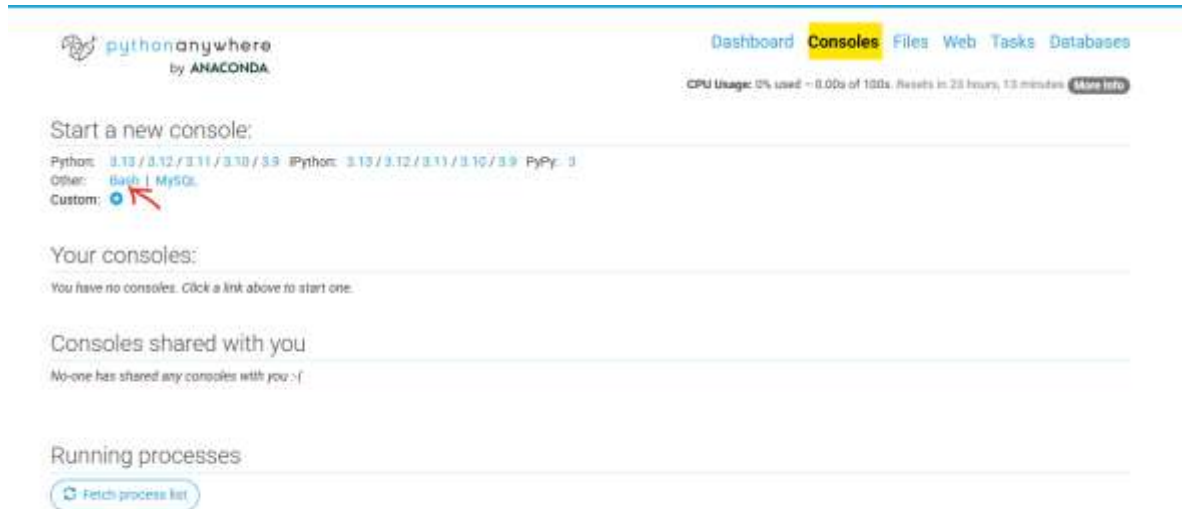
This month (previous month)	0 (0)
Today (yesterday)	0 (0)
Hour (previous hour)	0 (0)

Want some more data? Paying accounts get pretty charts :)

Code:



Abrir consola



Abrimos una consola bash.

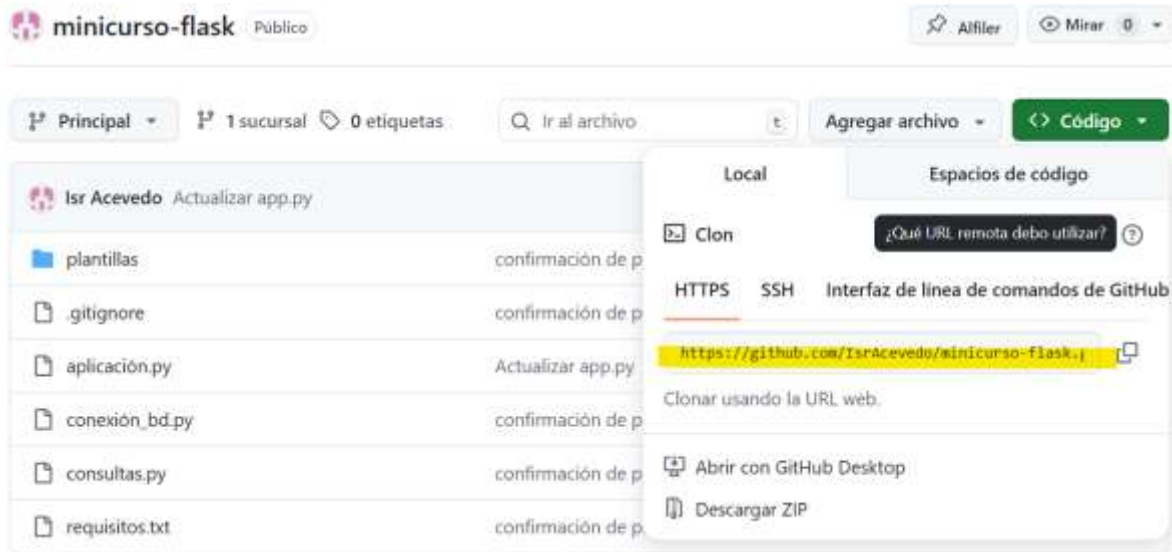


Recordamos algunos comandos vistos en clases anteriores.

pwd : me muestra la ruta en la cual estoy.

cd : permite abrir carpeta.

Lo que debemos hacer clonar el repositorio donde está el proyecto, como si estuviéramos en escritorio.



```
Bash console 42249010
15:36 ~ $ pwd
/home/sena25
15:39 ~ $ git clone https://github.com/IsrAcevedo/minicurso-flask.git
Cloning into 'minicurso-flask'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 14 (delta 2), reused 10 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (14/14), done.
Resolving deltas: 100% (2/2), done.
15:40 ~ $
```

Crear un entorno virtual

Activar el entorno virtual

```
Bash console 42249010
15:49 ~ $ python -m venv tienda
15:50 ~ $ source tienda/bin/activate
(tienda) 15:50 ~ $
```



Instalar todas las dependencias necesarias para el funcionamiento del proyecto. Terminada la instalación cerramos la consola.

```
Bash console 42240010
15:49 ~$ python -m venv tienda
15:50 ~$ source tienda/bin/activate
(tienda) 15:50 ~$ cd minicurso-flask
(tienda) 15:53 ~$ pip install -r requirements.txt
Looking in links: /usr/share/pip-wheels
Collecting blinker==1.9.0 (from -r requirements.txt (line 1))
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click==8.2.1 (from -r requirements.txt (line 2))
  Downloading click-8.2.1-py3-none-any.whl.metadata (2.5 kB)
Collecting colorama==0.4.6 (from -r requirements.txt (line 3))
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting Flask==3.1.2 (from -r requirements.txt (line 4))
  Downloading Flask-3.1.2-py3-none-any.whl.metadata (3.2 kB)
Collecting itsdangerous==2.2.0 (from -r requirements.txt (line 5))
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (3.9 kB)
Collecting Jinja2==3.1.6 (from -r requirements.txt (line 6))
  Downloading Jinja2-3.1.6-py3-none-any.whl.metadata (2.9 kB)
Collecting MarkupSafe==3.0.2 (from -r requirements.txt (line 7))
  Downloading MarkupSafe-3.0.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.0 kB)
Collecting mysql-connector-python==9.0.0 (from -r requirements.txt (line 8))
  Downloading mysql-connector-python-9.0.0-cp313-cp313-manylinux_2_28_x86_64.whl.metadata (7.5 kB)
Collecting python-dotenv==1.1.1 (from -r requirements.txt (line 9))
  Downloading python-dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Collecting werkzeug==3.1.3 (from -r requirements.txt (line 10))
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
  Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
  Downloading click-8.2.1-py3-none-any.whl (102 kB)
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
  Downloading Flask-3.1.2-py3-none-any.whl (103 kB)
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
  Downloading Jinja2-3.1.6-py3-none-any.whl (114 kB)
  Downloading MarkupSafe-3.0.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (23 kB)
  Downloading mysql-connector-python-9.0.0-cp313-cp313-manylinux_2_28_x86_64.whl (33.9 MB)
  Downloading python-dotenv-1.1.1-py3-none-any.whl (20 kB)
  Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Installing collected packages: python-dotenv, mysql-connector-python, MarkupSafe, itsdangerous, colorama, click, blinker, Werkzeug, Jinja2, Flask
```

Configurar la base de datos.

Vamos a exportar la base de datos que tenemos en local. Para eso usamos el comando `mysqldump` + las credenciales de la bd junto con el nombre de la bd y establecemos el nombre del archivo exportado > nombrearchivo.sql

```
C:\Users\EnginnerIsrael>mysqldump -u root -p prueba_flask > practicabd.sql
Enter password: ****

C:\Users\EnginnerIsrael>
```

En Pythonanywhere ir a la sección bases de datos, y crear una base de datos.

Para esto debemos establecer una contraseña segura.



Initialize MySQL

Let's get started! The first thing to do is to initialize a MySQL server:

Enter a new password in the form below, and note it down: you'll need it to access the databases once you've created them. You will only need to do this once.

New password:

.....

Confirm password:

.....

Initialize MySQL

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.

The screenshot shows the PythonAnywhere dashboard with the 'Databases' tab selected. A green notification bar at the top states 'Your MySQL service is now ready. Details below.' Below this, the 'MySQL' section is active, showing 'MySQL settings'. Under 'Connecting:', there is a red-bordered box containing the following information:

Database host address:	sema25.mysql.pythonanywhere-services.com
Username:	sema25

Below the box, a note states: 'Since you have a free account, you will only be able to connect to your database using code running inside PythonAnywhere.'

Luego ir archivos y en el archivo .env remplazar los valores por los datos definidos por la plataforma.



pythonanywhere by ANACONDA

Dashboard Consoles Files Web Tasks **Databases**

MySQL PostgreSQL

MySQL settings

Connecting
Use these settings in your web applications.

Database host address: `sema25.mysql.pythonanywhere-services.com` ✓
Username: `sema25` ✓

Since you have a free account, you will only be able to connect to your database using code running inside PythonAnywhere.

Your databases:
Click a database's name to start a MySQL console logged in to it.

Name
<code>sema25\$default</code>
<code>sema25\$practicatad</code> ✓

Create a database
Your database names always start with your username + '\$'. There's no need to type that prefix in below, though: PythonAnywhere will automatically add it.

Database name:

Create

El password fue el que definimos en el paso anterior.

pythonanywhere by ANACONDA

Dashboard Consoles **Files** Web Tasks Databases

Open Bash console here 29% full - 145.9 MB of your 512.0 MB quota [More info](#)

Directories

Enter new directory name

- ~/cache/
- ~/python/
- ~/local/
- ~/virtualenvs/
- ~/miniconda-flask/
- ~/sema25/

Files

Enter new file name, eg. hello.py

bash_history	1625-09-19 15:27	301 bytes
bashrc	2025-09-17 10:53	342 bytes
gitconfig	2025-09-17 10:53	284 bytes
my.cnf	2025-09-19 16:10	28 bytes
.profile	2025-09-17 10:53	74 bytes
pythonstartup.py	2025-09-17 10:53	77 bytes
.shrc	2025-09-17 10:53	46 bytes
README.txt	2025-09-17 10:53	132 bytes
<u>practicatad.sql</u>	2025-09-19 14:20	2.1 KB

Upload a file
100MB maximum size

Subimos el archivo .sql que exportamos anteriormente.

Abrimos nuevamente una consola, pero esta vez va a ser mysql.
esto ya nos abre con las credenciales. Por lo tanto, ya podemos trabajar las bases de datos como si estuviéramos en local.

Luego lo que hacemos es seleccionar la base de datos del servidor en la que vamos a trabajar e importamos la base de datos que teníamos en local y subimos en el paso anterior.



```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 58535444
Server version: 8.0.40 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
| sena25$default |
| sena25$practicabd |
+-----+
4 rows in set (0.36 sec)

mysql> USE sena25$practicabd;
Database changed
mysql> SHOW TABLES;
Empty set (0.01 sec)

mysql> source /home/sena25/practicabd.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.05 sec)

Query OK, 0 rows affected (0.00 sec)
```

Verificamos que se hayan cargado las tablas.



```
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_sena25$practicabd |
+-----+
| mensajes                     |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM mensajes;
+----+-----+-----+-----+
| id | nombre | celular | mensaje |
+----+-----+-----+-----+
| 1  | pepito | 31205562323 | hola, quiero aprender mas sobre flask |
| 2  | juanito | 3325265655 | Hola, yo tambein quiero aprender flask |
| 3  | juana  | 414545465 | excelente, me gusta lo que estoy aprendiendo |
| 4  | juana  | 414545465 | excelente, me gusta lo que estoy aprendiendo |
| 5  | juanita | 45532656 | todo perfecto |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> █
```

Configuración del WSGI

En la sección web, edita el archivo WSGI para cambiar la ruta donde está el proyecto clonado.



[/var/www/sena25_pythonanywhere_com_wsgi.py](#)

```
1 # This file contains the WSGI configuration required to serve up your
2 # web application at http://<your-username>.pythonanywhere.com/
3 # It works by setting the variable 'application' to a WSGI handler of some
4 # description.
5 #
6 # The below has been auto-generated for your Flask project
7
8 import sys
9
10 # add your project directory to the sys.path
11 project_home = '/home/sena25/minicurso-flask'
12 if project_home not in sys.path:
13     sys.path = [project_home] + sys.path
14
15 # import flask app but need to call it "application" for WSGI to work
16 from app import app as application # noqa
17
```

Agrega la ruta donde se encuentra el entorno virtual.



Python version: 3.13 

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/sena25/tienda`

[Start a console in this virtualenv](#)

Recarga la app desde el panel web en pythonanywhere

Clic en la url, y tu proyecto ya estará en producción.

