

SCRIPT DE LA BASE DE DATOS A UTILIZAR:

1. Abrir worbench y ejecutar este script en la instancia que crearon.

-- Crear la base de datos

```
CREATE DATABASE reposteria_db;
```

-- Usar la base de datos

```
USE reposteria_db;
```

-- Crear la tabla 'roles'

```
CREATE TABLE roles (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    role_name VARCHAR(50) NOT NULL UNIQUE  
);
```

-- Crear la tabla 'usuarios'

```
CREATE TABLE usuarios (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    role_id INT,  
    FOREIGN KEY (role_id) REFERENCES roles(id)  
);
```

-- Crear la tabla 'productos'

```
CREATE TABLE productos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,
```

```
price DECIMAL(10, 2) NOT NULL  
);
```

```
-- Insertar roles iniciales (opcional)
```

```
INSERT INTO roles (role_name) VALUES ('Administrador'), ('Vendedor'), ('Repostero');
```

2. Crear en vsc en la carpeta de proyecto la siguiente estructura:

Carpeta_proyecto:

```
|-----app/  
|-----__init__.py  
|-----routes/  
|-----auth.py  
|-----config.py  
|-----run.py  
|-----env  
|-----requirements.txt #este archivo lo generamos a través de la consola  
|-----env/
```

3. Codificando:

Archivo .env

```
SECRET_KEY=clave_muy_segura  
DB_HOST=localhost  
DB_USER=root  
DB_PASSWORD=''  
DB_NAME=repoteria_db
```

Archive config.py

```
import os  
from dotenv import load_dotenv
```

```
load_dotenv()

class Config:
    SECRET_KEY = os.getenv("SECRET_KEY")
    DB_HOST = os.getenv("DB_HOST")
    DB_USER = os.getenv("DB_USER")
    DB_PASSWORD = os.getenv("DB_PASSWORD")
    DB_NAME = os.getenv("DB_NAME")
```

Archive run.py:

```
from app import create_app

app = create_app()

if __name__ == '__main__':
    app.run(debug=True)
```

archive auth.py

```
# app/routes/auth.py
from flask import Blueprint

auth_bp = Blueprint('auth', __name__)

@auth_bp.route('/')
def home():
    return "Bienvenido al sistema de repostería"
```

archivo __init__.py

```
from flask import Flask
from config import Config
import mysql.connector

def create_app():
    app = Flask(__name__)
    app.config.from_object(Config)

    app.config['MYSQL_CONNECTION'] = mysql.connector.connect(
        host=Config.DB_HOST,
```

```
        user=Config.DB_USER,  
        password=Config.DB_PASSWORD,  
        database=Config.DB_NAME  
    )  
  
    from app.routes.auth import auth_bp  
    app.register_blueprint(auth_bp)  
  
    return app
```

4. En la terminal
 1. Activar el entorno
 2. Colocar el siguiente comando:
pip install flask python-dotenv mysql-connector-python
 3. Creamos el archive de requirements.txt, a través del siguiente comando:
pip freeze > requirements.txt
 4. Ejecutamos el proyecto con el comando:
python run.py