

Lección 2: Actividad de portafolio nro. 2

“Implementación de un modelo de usuarios con Django ORM y PostgreSQL”

Nicolás Jofré Andrade - Desarrollo Full-stack Python
05-2024

Objetivo: Crear un proyecto Django que utilice el ORM para gestionar usuarios en una base de datos con Postgres, realizando operaciones CRUD (crear, leer, actualizar, borrar) con el modelo de usuario.

1. Instalación de dependencias:

- Instala el paquete **psycopg2** para la conexión con PostgreSQL.

```
TERMINAL

• (venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod7\actividad2> pip install psycopg2
Requirement already satisfied: psycopg2 in c:\users\divalcl\desktop\dev\portafolio-py-mod7\actividad2\venv\lib\site-packages (2.9.9)
○ (venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod7\actividad2> █
```

```
TERMINAL

• (venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod7\actividad2> pip freeze
asgiref==3.8.1
Django==5.0.6
psycopg2==2.9.9
sqlparse==0.5.0
tzdata==2024.1
```

2. Creación del proyecto Django y configuración de la base de datos:

- Crea un nuevo proyecto Django llamado ‘UsuariosProject’.

```
• (venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod7\actividad2> django-admin startproject usuariosProject
○ (venv) PS C:\Users\DivalCL\Desktop\dev\portafolio-py-mod7\actividad2> █
```

- Configura la base de datos en ‘settings.py’. Utilizaremos PostgreSQL, así que asegúrate de tener un servidor PostgreSQL instalado y crea una nueva base de datos.

3. Definición de un modelo de usuario:

- En la aplicación ‘usuarios’, crea un modelo de usuario en ‘models.py’.

4. Operaciones CRUD con el modelo:

- Ejecuta las migraciones para aplicar el modelo a la base de datos.
- Abre el shell interactivo de Django y realiza operaciones CRUD con el modelo.