

DRILLING MODULO 6 SESION 2 SR YURI URZUA LEBUY

Los pasos a seguir son los siguientes:

1. Crear el proyecto en Django con el nombre de site_django, dentro de la carpeta practica_django.

```
D:\0001 BOOT CAMP\BOOTCAMP\M6\SESION2\practica_django>django-admin startproject site_django
```

2. Abrir con Visual Studio Code los archivos creados.

```
code .
```

3. Crear una nueva aplicación en el proyecto Django llamada Book (Libro), y agregarla al proyecto site_django.

```
D:\0001 BOOT CAMP\BOOTCAMP\M6\SESION2\practica_django\site_django>python manage.py startapp Book
```

4. Ejecutar el servidor de desarrollo de Django.

```
D:\0001 BOOT CAMP\BOOTCAMP\M6\SESION2\practica_django\site_django>python manage.py runserver
```

```
Watching for file changes with StatReloader
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

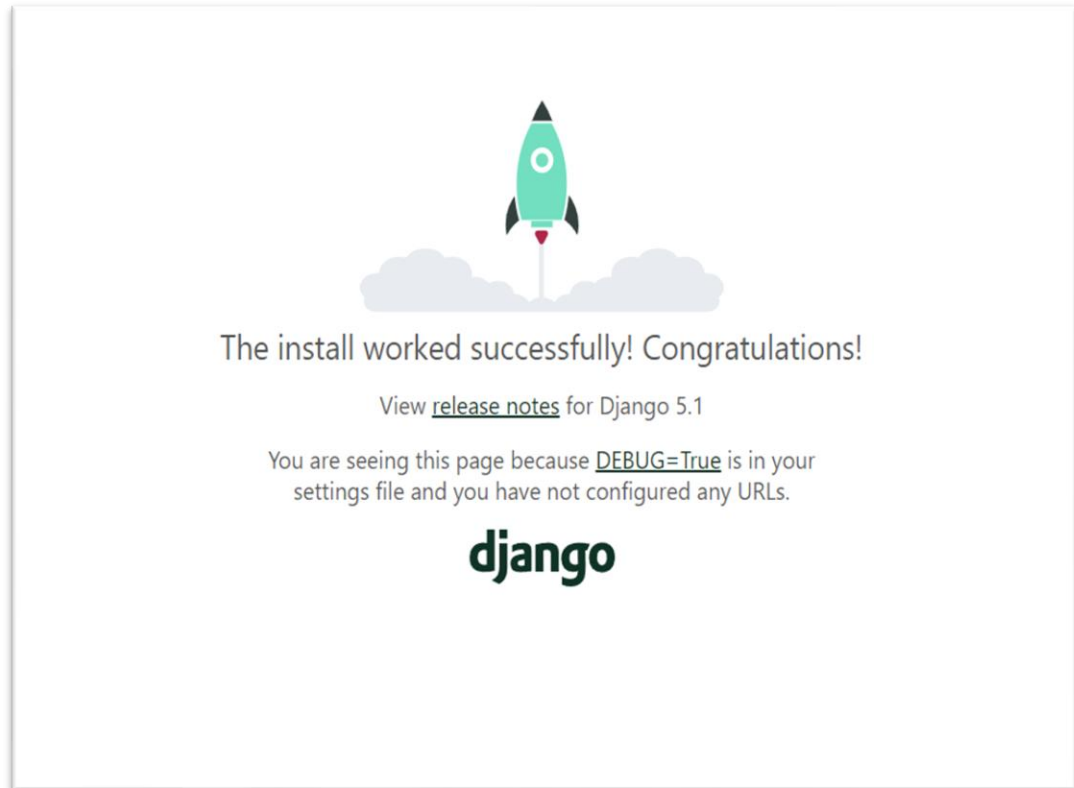
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions. Run 'python manage.py migrate' to apply them.

November 24, 2024 - 16:25:29

Django version 5.1.3, using settings 'site_django.settings'

Starting development server at <http://127.0.0.1:8000/>

Quit the server with CTRL-BREAK.



5. Con la ayuda de la terminal, describa qué función realiza cada uno de los siguientes comandos:

makemigrations: Crea archivos de migración basados en los cambios detectados en los modelos. Estos archivos contienen las instrucciones para modificar la base de datos.

migrate: Aplica las migraciones pendientes a la base de datos. Ejecuta los cambios definidos en los archivos de migración para actualizar el esquema de la base de datos.

shell: Inicia una consola interactiva de Python con el contexto de Django cargado. Permite interactuar con los modelos y el ORM directamente.

dbshell: Abre una consola interactiva de la base de datos configurada (ej: SQLite, PostgreSQL). Permite ejecutar consultas SQL directamente.

showmigrations: Muestra el estado de todas las migraciones del proyecto, indicando cuáles se han aplicado y cuáles están pendientes.

dumpdata: Exporta los datos de la base de datos a un archivo en formato JSON, XML o YAML. Útil para crear copias de seguridad o fixtures.

test: Ejecuta las pruebas automatizadas definidas en el proyecto.

testserver: Inicia un servidor de desarrollo usando una base de datos temporal, cargando los datos desde un fixture especificado.

diffsettings: Muestra las diferencias entre la configuración actual y la configuración por defecto de Django. Útil para ver qué configuraciones se han modificado.