**Ruta del repositorio:**

https://github.com/Freston1605/python-individual

**Respuestas a las preguntas del ejercicio:**

Paquetes instalados en el entorno virtual:

Comando: pip list (con el entorno activado)

Package Version

---------------- --------

asgiref 3.7.2

certifi 2023.5.7

distlib 0.3.6

Django 4.2.2

filelock 3.12.0

pip 23.1.2

pipenv 2023.6.2

platformdirs 3.5.1

setuptools 67.8.0

sqlparse 0.4.4

tzdata 2023.3

virtualenv 20.23.0

virtualenv-clone 0.5.7

Ventajas de usar Django:

1. **Arquitectura MVC (Modelo-Vista-Controlador):** Django sigue el patrón de diseño MVC, lo que facilita la separación de la lógica de negocios, la presentación y la interacción con la base de datos.
2. **ORM (Object-Relational Mapping):** Django incluye un ORM potente que permite interactuar con la base de datos utilizando objetos Python en lugar de escribir consultas SQL directamente. Esto simplifica el acceso y la manipulación de datos.
3. **Administrador de Django:** Proporciona una interfaz administrativa preconstruida y personalizable que permite gestionar los modelos de datos de manera sencilla. Esto incluye tareas como la creación, lectura, actualización y eliminación (CRUD) de registros de la base de datos.
4. **Enrutamiento de URLs:** Django ofrece un sistema de enrutamiento de URLs fácil de usar que permite mapear las URL de tu aplicación a vistas específicas. Esto facilita la navegación y la construcción de URLs coherentes en tu aplicación.
5. **Plantillas (Templates):** Django incluye un sistema de plantillas que permite separar la lógica de presentación del código Python. Esto facilita la creación de interfaces de usuario dinámicas y reutilizables.
6. **Seguridad integrada:** Django ofrece funciones de seguridad integradas para proteger tu aplicación contra vulnerabilidades comunes, como ataques de inyección de SQL, ataques de falsificación de solicitudes entre sitios (CSRF) y ataques de scripting entre sitios (XSS).
7. **Autenticación y autorización:** Django proporciona un sistema de autenticación y autorización robusto que facilita la gestión de usuarios, el inicio de sesión, el registro y la gestión de permisos y roles.
8. **Internacionalización y localización:** Django ofrece soporte completo para la internacionalización y localización de aplicaciones web, lo que facilita la traducción de tu aplicación a diferentes idiomas y la adaptación a diferentes regiones.
9. **Caché y optimización de consultas:** Django incluye mecanismos de caché integrados que ayudan a mejorar el rendimiento de tu aplicación al almacenar en caché resultados de consultas costosas. También proporciona herramientas para optimizar consultas de base de datos y reducir la carga en el servidor.
10. **Gestión de formularios:** Django facilita la creación y validación de formularios HTML, así como el manejo de datos enviados por el usuario.

¿Es posible hacer un servidor en Python sin django?

Sí, es posible. Algunas de las alternativas es usar otros entornos de trabajo como flask o módulos como http.server.

¿Cuáles son las desventajas de usar otros métodos?

No usar django nos trae desventajas como una menor cantidad de funciones integradas por defecto, una curva de aprendizaje más pronunciada, menor soporte al tener una comunidad menor, menor seguridad integrada a la plataforma y soluciones subóptimas.