Servidor Falso de DOTERS

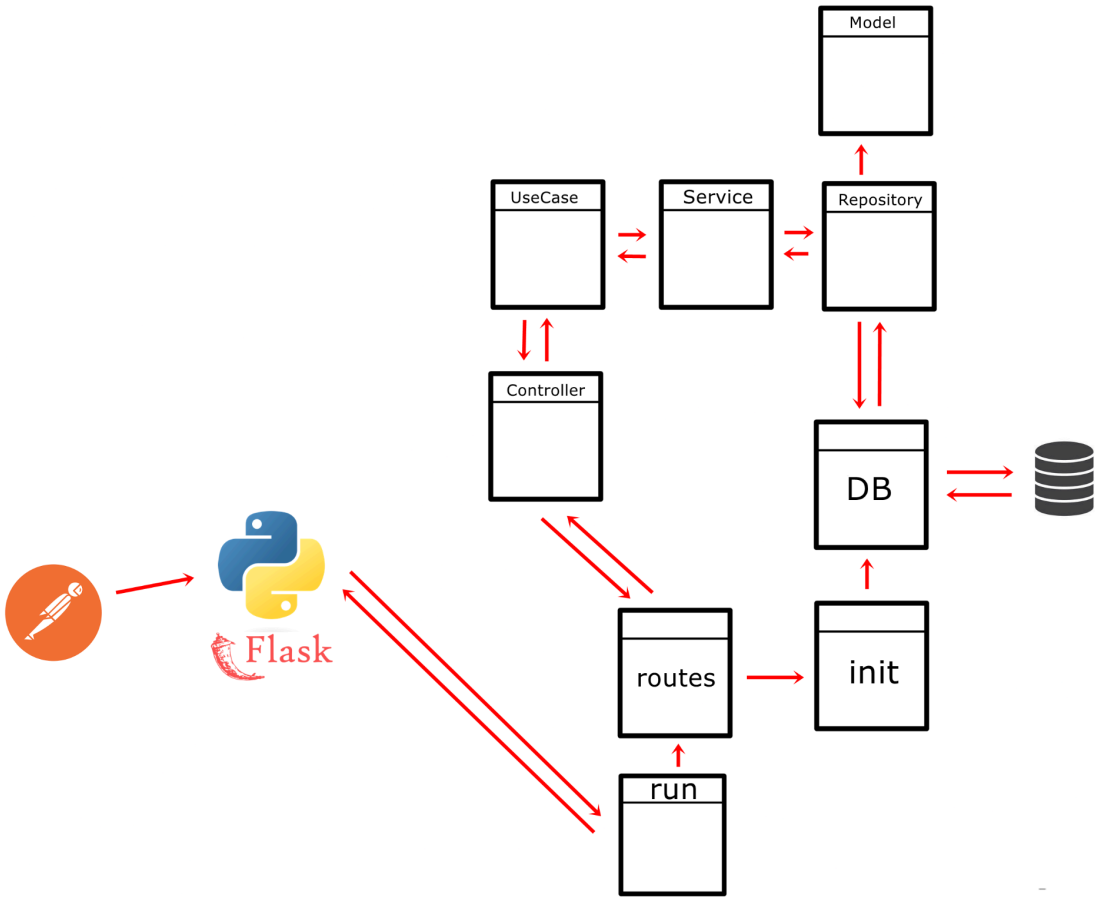
Para poder continuar con nuestro desarrollo vamos a tener un servidor FLASK en python el cual va a simular todas las respuestas… para ello vamos a tener la siguiente arquitectura:

Mock server en Flask que emula endpoints de Doters. Diseño modular y explícito para crecer por casos de uso (signup, auth, points, etc.).

Capa	**Responsabilidad**	**Descripción**
Routing	app/routes.py	Registra blueprints o handlers por dominio.
INIT	app/__init__.py	Inicializa la APP, ROUTER y DB.
Helpers	app\helpers\response.py	Genera una respuesta estándar JSON.
Controller	app/controllers/*	Capa de orquestación HTTP.
UseCase	app/UseCases/*	Orquesta servicios para resolver la lógica de negocio.
Services	app/services/*	Es el código de lógica de negocio.
Repositorios	app/repositories/*	Realiza llamados a DB usando Modelos. Acceso a datos (SQLite) usando `DbContext`.
Modelos	app/models/*	Son las entidades del dominio de negocio.

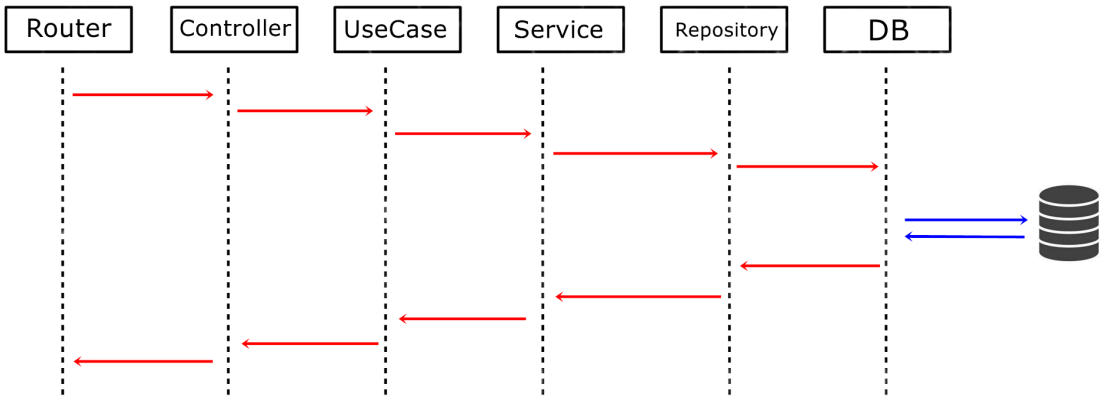
Infraestructura	app\Database\context.py	Singleton de SQLite, crea la conexión a DB y expone la base de datos.

Diagrama de alto nivel



ROUTER > CONTROLLER > UseCase > Service > Repository > DB

Diagrama de secuencia



Como crear un nuevo ROUTER FULL

- Crear model en caso de que no exista:
  app/models
- Crear la tabla de DB en caso de que no exista:
  app/Database/context.py
- Crear el repositorio:
  app/repositories
- Crear el servicio:
  app/services
- Crear el caso de uso:
  app/UseCases
- Crear el controlador:
  app/controllers
- Registrar en el router:
  app/routes.py

Ejemplo para crear la feature login

Paso 0:
Verificar la existencia de la tabla usuario que contenga email y pass en DB:
app\Database\context.py

Paso 1:
Crear el repositorio que va a extraer email y pass de la tabla User:
app\repositories\login_repository.py

Paso 2:
Crear el servicio que se encargará de hacer la llamada al el repositorio:
app/services/login_service.py

Paso 3:
Crear el caso de uso el cual ejecutará el request para dar el resultado:

app/UseCases/login_use_case.py