**Sistema de Misiones RPG con Colas**

Este proyecto implementa un sistema de misiones para un juego RPG utilizando una estructura de datos (TDA Cola) para gestionar el orden FIFO de las misiones, un ORM (SQLAlchemy) para la persistencia de datos y una API REST con FastAPI para interactuar con el sistema.

**Características**

* **TDA Cola (FIFO):**  
  Implementación en tda\_cola.py con métodos:
  + enqueue(mission): Añadir misión al final.
  + dequeue(): Eliminar/retornar la primera misión.
  + first(): Ver la primera misión sin remover.
  + is\_empty(): Verificar si está vacía.
  + size(): Obtener la cantidad de misiones.
* **ORM:**  
  Utiliza SQLAlchemy para mapear clases de Python a tablas de la base de datos.
  + Los modelos se definen en models.py (clases Mision, Personaje y MisionPersonaje).
  + La conexión y creación de la base de datos se configura en database.py.
* **API REST:**  
  Desarrollada con FastAPI en app.py.
  + Permite crear personajes y misiones.
  + Permite asignar misiones a personajes (integrando la TDA Cola).
  + Permite completar misiones (por misión y personaje).
  + Permite listar las misiones asignadas a cada personaje, mostrando también el nombre del personaje.

**Requisitos**

* Python 3.x
* Dependencias listadas en requirements.txt:
  + fastapi
  + uvicorn
  + sqlalchemy
  + pydantic

**Instalación**

1. Clona el repositorio:

bash

CopiarEditar

git clone <URL-del-repositorio>

1. Navega a la carpeta del proyecto:

bash

CopiarEditar

cd rpg\_misiones

1. Instala las dependencias:

bash

CopiarEditar

pip install -r requirements.txt

**Configuración de la Base de Datos**

Para crear la base de datos y las tablas, ejecuta:

bash

CopiarEditar

python database.py

Esto generará el archivo rpg\_misiones.db con las tablas necesarias.

**Ejecución de la API**

Inicia el servidor FastAPI con Uvicorn:

bash

CopiarEditar

uvicorn app:app --reload

La API estará disponible en: <http://127.0.0.1:8000>

Puedes acceder a la documentación interactiva de la API en: http://127.0.0.1:8000/docs

**Endpoints Principales**

* **Crear Personaje:**  
  POST /personajes  
  Parámetro: nombre (string)
* **Crear Misión:**  
  POST /misiones  
  Parámetros: nombre (string), descripcion (string), experiencia (integer)
* **Aceptar Misión:**  
  POST /personajes/{personaje\_id}/misiones/{mision\_id}  
  Asigna una misión a un personaje y la encola (TDA Cola).
* **Completar Misión:**  
  POST /personajes/{personaje\_id}/completar/{mision\_id}  
  Completa una misión específica y actualiza la experiencia del personaje.
* **Listar Misiones del Personaje:**  
  GET /personajes/{personaje\_id}/misiones  
  Muestra el nombre del personaje y las misiones asignadas en orden FIFO.