DOCUMENTO TÉCNICO DEL PROYECTO

1. Introducción

Este documento describe la arquitectura, componentes, modelos, rutas, vistas HTML y funcionalidades del proyecto NPC basado en FastAPI. Su propósito es detallar el funcionamiento interno del sistema y evidenciar el cumplimiento de los requisitos académicos.

2. Tecnologías Utilizadas

- FastAPI como framework principal.

- SQLModel para modelos y ORM.

- SQLite como base de datos.

- Jinja2 para renderizado de plantillas HTML.

- HTML, CSS y formularios para el frontend.

- Uvicorn para despliegue local.

- Estructura modular en routers y servicios.

3. Arquitectura del Proyecto

El proyecto sigue una arquitectura organizada en carpetas:

- main.py: punto de entrada de la aplicación.

- routers/: contiene los endpoints agrupados por entidad (npcs, items, misiones, ubicaciones, búsqueda, reportes).

- modelos/: define las clases SQLModel que representan las tablas de la base de datos.

- servicios/: implementa lógica de negocio independiente de las rutas.

- templates/: incluye vistas HTML como formularios, listados y detalles.

- static/: incluye CSS y recursos estáticos.

4. Modelos Implementados

El proyecto cuenta con múltiples modelos que representan entidades del dominio:

- NPC

- Item

- Misión

- Ubicación

- Tablas relacionales (NPCItemLink, etc.)

Estos modelos cuentan con relaciones y permiten implementar operaciones CRUD completas.

5. Endpoints y Funcionalidades

Se implementan rutas para:

- Crear, listar, actualizar y eliminar NPCs.

- CRUD de Items.

- CRUD de Misiones.

- CRUD de Ubicaciones.

- Búsqueda dentro del sistema mediante parámetros.

6. Frontend y Formularios HTML

El proyecto incluye:

- Formularios de ingreso de datos.

- Páginas de detalles para cada entidad.

- Sistema de navegación.

- Aplicación de estilos CSS.

- Búsqueda desde HTML.

7. Validación y Reglas de Negocio

- Validación realizada mediante SQLModel y FastAPI.

- Validación en formularios HTML.

- Lógica de negocio implementada en servicios independientes de los endpoints.

8. Despliegue

El proyecto está preparado para ser desplegado usando:

- Render, Railway, o cualquier servicio compatible con Uvicorn.

El archivo main.py expone la app "FastAPI()" siguiendo los lineamientos para despliegue.

9. Conclusiones

El proyecto cumple con las características solicitadas:

- Uso de FastAPI como base central.
- Implementación de al menos dos modelos con relaciones.
- CRUD completo.
- HTML navegable y con estilos.
- Formularios funcionales.
- Búsqueda y validación.