**Resumen del Proyecto**

Gestor de Evidencias MP es un sistema de gestión integral para el Ministerio Público que facilita el registro, asignación, trazabilidad y reporte de evidencias forenses. El objetivo es optimizar los procesos internos, garantizar seguridad y trazabilidad en la cadena de custodia y ofrecer herramientas de consulta y generación de reportes en formatos estándar (CSV y PDF).

**Tecnologías Usadas y Sus Usos**

* **Python 3.10+ con Django 4.x**: Framework backend para la implementación de la lógica de negocio, modelos de datos y APIs REST.
* **Django REST Framework**: Extensión de Django para construir APIs REST robustas y con interfaz "browsable" para desarrolladores.
* **PostgreSQL**: Motor de base de datos relacional para almacenamiento ACID de evidencias y registros de auditoría.
* **Docker & Docker Compose**: Contenerización de la aplicación y orquestación de servicios (web, db, backup) para entornos reproducibles.
* **React 19.x**: Librería frontend basada en componentes para crear interfaces de usuario interactivas y dinámicas.
* **Tailwind CSS 4.x**: Framework CSS "utility-first" para estilos rápidos y consistentes en el frontend.
* **Git & GitHub Actions**: Control de versiones distribuido y pipeline de CI/CD para automatizar tests, builds y despliegues.
* **pytest**: Framework de testing para Python que facilita la creación de pruebas unitarias y funcionales.

**Plan Semanal de Tareas (8 Semanas)**

| **Semana** | **Tareas Principales** |
| --- | --- |
| 1 | - Configurar repositorio Git y ramas. |
|  | - Crear README.md, requirements.txt, .gitignore, Dockerfile, docker-compose.yml. |
|  | - Definir convenciones de código y crear .cursorrules. |
| 2 | - Implementar autenticación de usuarios: modelo extendido y endpoints login/logout. |
|  | - Configurar roles: Administrativo, Operativo, Administrador. |
|  | - Escribir tests de autenticación (pytest). |
| 3 | - CRUD de usuarios (crear, editar, eliminar, bloquear, reasignar roles). |
|  | - Encriptación de contraseñas y manejo de sesiones seguras. |
|  | - Tests de permisos por rol. |
| 4 | - Modelo y formulario de Registro de Evidencias con todos los campos requeridos. |
|  | - Lógica de asignación automática a expertos y registro de historial. |
|  | - Tests de validación de datos y asignaciones. |
| 5 | - Desarrollo de interfaz: tabla estilo Excel para evidencias con filtros y paginación. |
|  | - Prototipos UI con React y Tailwind. |
| 6 | - Formulario de Salida de Evidencias y descarga de PDF de cadena de custodia. |
|  | - Endpoints para generación de reportes CSV y PDF. |
|  | - Tests de integración end-to-end. |
| 7 | - Implementar módulo de auditoría: visualización de logs para Administrador. |
|  | - Configurar backups automáticos diarios en contenedor DB. |
|  | - Pruebas de carga y rendimiento (consulta ≤10,000 registros). |
| 8 | - Corrección de bugs y refinamiento según feedback. |
|  | - Documentación final de usuario y despliegue (docs/). |
|  | - Configurar pipeline CI/CD y desplegar en staging. |

Este documento sirve como guía inicial para el desarrollo y despliegue del sistema, detallando objetivos, tecnologías y tareas semanales claras para el equipo de trabajo.