

PFO 2: Sistema de Gestión de Tareas con API y Base de Datos

Al finalizar este trabajo práctico, serás capaz de:

1. Implementar una **API REST** con endpoints funcionales.
2. Utilizar **autenticación básica** con protección de contraseñas.
3. Gestionar datos persistentes con **SQLite**.
4. Construir un **cliente en consola** que interactúe con la API.

Servidor (API Flask)

Desarrolla un servidor que realiza lo siguiente:

1. **Registro de Usuarios**
 - Endpoint: POST /registro
 - Debe recibir {"usuario": "nombre", "contraseña": "1234"}.
 - Almacenar usuarios en SQLite con contraseñas hasheadas (¡nunca en texto plano!).
2. **Inicio de Sesión**
 - Endpoint: POST /login
 - Verifica credenciales y permite acceso a las tareas.
3. **Gestión de Tareas**
 - GET /tareas: Muestre un html de bienvenida

Requisitos técnicos:

- Usar alguna librería para hashear contraseñas.
- La persistencia de los datos debe ser en sqlite.

Entregables

1. **Código Fuente:**
 - servidor.py (API Flask + SQLite).
2. **Documentación:**
 - Instrucciones para ejecutar el proyecto y probarlo(README.md).

- Capturas de pantalla de pruebas exitosas.
- Entregable en repositorio de Github, usando Github pages alojar el proyecto.

3. Respuestas Conceptuales:

- ¿Por qué hashear contraseñas?
- Ventajas de usar SQLite en este proyecto.