

Informe – Mini Ciclo de Desarrollo con IA + SQLite + GitHub

1. Planificación

Objetivo del sistema: Construir una aplicación de gestión de tareas (To-Do List) que permita:

- Agregar nuevas tareas.
- Corregir (editar) tareas existentes.
- Marcar tareas como completadas.
- Eliminar tareas (individuales o todas).
- Mostrar (listar) todas las tareas almacenadas en la base de datos.

Base de datos definida:

- Nombre: tasks.db
- Tabla: tasks con la siguiente estructura:
id (INTEGER, clave primaria, autoincremental)
title (TEXT, título de la tarea)
status (TEXT, 'pendiente' o 'completada')

2. Prototipo con IA

IA utilizada: ChatGPT (GPT-5).

Prompt inicial usado: "Crea un programa en Python que gestione una lista de tareas con SQLite.

La base de datos debe llamarse tasks.db y la tabla tasks con columnas id, title, status.

Implementa funciones para agregar, listar y marcar como completada, usando interacción desde consola."

Código inicial generado: versión en consola en Python con SQLite.

Explicación: La IA generó un programa que crea automáticamente tasks.db, con persistencia de datos,

y permite manejar tareas básicas desde la terminal.

3. Prueba y Ejecución (versión consola)

Se realizaron pruebas agregando dos tareas, listando y marcando una como completada.

El sistema funcionó correctamente en consola, aunque con interfaz básica.

4. Retroalimentación y Mejora

Problemas detectados:

- El menú de consola era poco intuitivo.
- El listado de tareas no se veía claramente como "reporte".

Nuevo prompt a la IA: "Convierte el sistema de tareas en una aplicación con ventanas usando Tkinter.

Agrega botones para agregar, listar, editar, borrar y marcar como completada. Haz que el botón 'Listar todas las tareas' abra un popup con el listado completo desde la base de datos."

Código mejorado: versión gráfica con Tkinter y popup de listado.

Explicación: Ahora el sistema muestra tareas en una ventana gráfica, permite editar y borrar con más facilidad, y el botón "Listar todas las tareas" abre un popup con todas las tareas guardadas.

5. Versionamiento con GitHub

Se utilizó GitHub para control de versiones, con los siguientes comandos:

```
git init
git add .
git commit -m "Versión Tkinter con popup para listar tareas"
git branch -M main
git remote add origin https://github.com/TU-USUARIO/todo_app.git
git push -u origin main
```

6. Conclusiones

1. La IA facilitó la creación del prototipo inicial en consola y su conversión a interfaz gráfica con Tkinter.
2. Se cumplió el ciclo de desarrollo: planificación → prototipo → prueba → feedback → mejora → versionamiento.
3. SQLite garantizó persistencia real de los datos en tasks.db.
4. El popup de "Listar todas las tareas" asegura el cumplimiento del requerimiento de listado.
5. GitHub permitió mantener historial de cambios y entregar el proyecto de forma profesional.

7. Anexos

Se recomienda añadir capturas de pantalla en el documento Word o presentación final, tales como:

- Carpeta mostrando app.py y tasks.db.
- Ejecución en consola (versión inicial).
- Ventana Tkinter funcionando.
- Popup con listado de tareas.
- Repositorio en GitHub con los archivos subidos.