Gestión de Tareas con Java, JUnit y GitHub

# Introducción

Una pequeña empresa necesita organizar mejor las tareas pendientes de su equipo de trabajo. Hasta ahora, los encargados de proyectos se manejaban con notas sueltas o recordatorios dispersos, lo cual dificultaba el seguimiento del progreso. Con este proyecto se desarrolla una aplicación sencilla en Java que funcione como un gestor de tareas.

# Objetivos

- Permitir registrar nuevas tareas.  
- Marcar tareas como completadas.  
- Listar todas las tareas mostrando su estado.  
- Eliminar tareas innecesarias.  
- Contar con un menú interactivo en consola.  
- Validar el funcionamiento mediante pruebas unitarias con JUnit.  
- Gestionar el código con Git y subirlo a un repositorio en GitHub.

# Desarrollo del Proyecto

## 1. Lógica Principal

Se implementa la clase GestorTareas, que administra una lista de objetos de tipo Tarea. La clase ofrece métodos para agregar, completar, eliminar y listar tareas.

## 2. Interfaz en Consola

Se diseña un menú interactivo que permite al usuario acceder a las distintas funcionalidades del gestor de manera ordenada desde la terminal.

## 3. Pruebas Unitarias con JUnit

Se implementan pruebas unitarias que validan los casos fundamentales:  
- Que al añadir una tarea quede registrada.  
- Que se pueda cambiar su estado de pendiente a completada.  
- Que al eliminar una tarea ya no aparezca en el listado.

Texto

El contenido generado por IA puede ser incorrecto.

## 4. Control de Versiones con Git y GitHub

Durante el desarrollo se utilizan commits para registrar los avances. El proyecto se sube a un repositorio en GitHub, cuyo historial refleja claramente la secuencia de cambios.

# Evidencias

A continuación se incluyen capturas de pantalla de la ejecución del programa, las pruebas unitarias exitosas y el repositorio en GitHub.

# Repositorio en GitHub

URL del repositorio: <https://github.com/Pablogg25/PruebaDeConocimiento-Dvops.git>

# Conclusión

El proyecto permitió crear una aplicación de consola en Java que gestiona tareas de forma sencilla y eficaz. Las pruebas unitarias garantizaron la corrección de la lógica y GitHub facilitó la colaboración y el control de versiones.