

Principios SOLID

Nombre: Manuel Valdes

Lenguaje: Java

Objetivo del Proyecto

Desarrollar un sistema modular y mantenible para gestionar tareas, proyectos y usuarios. La arquitectura implementa de forma explícita los cinco principios SOLID para mejorar la cohesión, evitar dependencias rígidas y facilitar la escalabilidad del código.

Clases e Interfaces Principales

Clase / Interfaz	Descripción	Principios aplicados
Task	Representa una tarea individual. Tiene título, descripción y estado.	S
User	Representa un usuario con lista de tareas asignadas.	S
Project	Representa un proyecto que contiene tareas.	S
TaskManager	Administra tareas y su asignación. Depende de interfaces, no de clases concretas.	D, I
ITareaManager	Interfaz para manejar tareas.	I
IAsignacionManager	Interfaz para asignar tareas a usuarios o proyectos.	I
UsuarioAdmin	Extiende User y puede marcar todas las tareas como completadas.	L
TaskPrinter	Clase adicional para imprimir tareas. Extendible sin modificar lógica existente.	O
Main	Contiene la interfaz de usuario por consola.	–

Aplicación de los Principios SOLID

- **S - Single Responsibility**

Cada clase tiene una sola responsabilidad:

Task: manejar estado y datos de la tarea.

User: manejar asignación de tareas al usuario.

Project: manejar lista de tareas de un proyecto.

- **O - Open/Closed**

La clase TaskPrinter permite extender la visualización de tareas sin modificar la lógica de TaskManager. Si se requiere imprimir en HTML o PDF, se puede heredar o sobrescribir.

- **L - Liskov Substitution**

UsuarioAdmin es una subclase válida de User, y puede sustituirla sin romper la funcionalidad esperada, cumpliendo con la herencia segura.

- **I - Interface Segregation**

Se crean interfaces pequeñas y específicas:

ITareaManager solo gestiona tareas.

IAsignacionManager solo se encarga de asignaciones.

Esto evita que una clase implemente métodos que no necesita.

- **D - Dependency Inversion**

TaskManager depende de interfaces (ITareaManager, IAsignacionManager) en lugar de clases concretas, lo que permite mayor flexibilidad y facilidad de pruebas.

Interfaz de Usuario

- Basada en consola.
- Menú interactivo con opciones para:
 - Crear tareas, usuarios y proyectos.
 - Asignar tareas a usuarios o proyectos.
 - Ver el listado de tareas registradas.

Pruebas y Validación

Se han realizado pruebas funcionales manuales mediante el menú en consola. Casos evaluados:

- Agregar y visualizar tareas.
- Asignación de tareas a usuarios y proyectos.
- Ejecución de comportamiento extendido desde UsuarioAdmin.

Conclusión

Este sistema demuestra cómo una arquitectura basada en SOLID permite:

- Mejor mantenimiento y legibilidad del código.
- Bajo acoplamiento y alta cohesión.
- Facilidad de escalabilidad (nuevas funcionalidades sin modificar código existente).
- Facilidad de pruebas gracias a la inversión de dependencias.