**Documentación de tarea de principios SOLID**

**Estructura del Sistema**

Clases:

Task: Representa una tarea.

Project: Representa un proyecto que contiene varias tareas.

User: Representa un usuario que puede trabajar en tareas.

TaskManager: Maneja la lógica de gestión de tareas.

Interfaces:

TaskIfc: Interfaz para las operaciones de las tareas.

ProjectIfc: Interfaz para las operaciones de los proyectos.

UserIfc: Interfaz para las operaciones de los usuarios.

TaskManagerIfc: Interfaz para las operaciones del gestor de tareas.

**Explicación de la Implementación**

Single Responsibility Principle (SRP):

Cada clase tiene una única responsabilidad. Por ejemplo, Task maneja la lógica de   
 las tareas, Project se encarga de la gestión de tareas dentro de un proyecto y User   
 representa a los usuarios.

Open/Closed Principle (OCP):

Las clases están diseñadas para ser extensibles sin necesidad de modificar el código   
 existente.

Si se desea agregar una nueva tarea o tipo de usuario, se pueden crear nuevas   
 clases que extiendan la funcionalidad.

Liskov Substitution Principle (LSP):

Cualquier instancia de TaskIfc, ProjectIfc, o UserIfc puede ser reemplazada por sus   
 implementaciones concretas sin afectar la funcionalidad del sistema.

Interface Segregation Principle (ISP):

Se han creado interfaces específicas para cada clase, evitando que una clase   
 implemente métodos que no necesita.

Dependency Inversion Principle (DIP):

Las clases dependen de interfaces en lugar de implementaciones concretas. Esto   
 facilita la inyección de dependencias y mejora la testabilidad.