

Eduardo Felipe Poot Chairez

PATRONES DE DISEÑO

¿Qué es?

Los patrones de diseño son soluciones generales y reutilizables a problemas comunes que aparecen de forma repetida al diseñar software.

Son modelos o guías que indican cómo estructurar clases, objetos y sus relaciones para resolver un problema de manera efectiva.

Puntos a considerar

- Se pueden combinar muchos patrones de diseño.
- Los patrones son independientes del lenguaje de programación y de la tecnología.

Utilidad

- Ayudan a definir los módulos a usar.
- Indica qué función tienen cada módulo.
- Define como se relaciona cada módulo (herencia, invocación de servicios, creación de otros módulos)

Tipos de patrones

- **Creacionales:** Se enfocan en cómo se instancian las clases.
- **Estructurales:** Se enfocan en cómo se relacionan las clases.
- **De comportamiento:** Se enfocan en cómo interactúan los objetos.

Singleton (Creacional)

Es un patrón que garantiza que solo exista **UNA** instancia de una clase en todo el programa.

Ejemplo:

La conexión con una base de datos.

Adapter (Estructural)

Permite que clases incompatibles trabajen juntas.

Ejemplo:

El uso de una librería vieja o externa y la necesidad de que funcione con tu código actual.

Observer (De comportamiento)

Permite que varios objetos sean notificados automáticamente cuando otro cambia.

Ejemplo:

Eventos, interfaces gráficas, sistemas de notificaciones, listeners.

Beneficios

- **Ahorro de tiempo.**
- **Calidad:** Los diseños son comúnmente probados por expertos.
- **Vocabulario común:** Permite que un equipo de desarrolladores hable el mismo idioma