

PATRÓN DE DISEÑO

1

¿QUÉ ES UN PATRÓN DE DISEÑO?

Los patrones de diseño son soluciones comunes a problemas repetitivos en el desarrollo de software. Son como guías que muestran cómo resolver situaciones en programación de una forma organizada.

2

ESTRUCTURA DE UN PATRÓN

- Propósito: Describe el problema y la solución general.
- Motivación: Explica en más detalle el problema y cómo el patrón ayuda a resolverlo.
- Estructura de Clases: Muestra las partes del patrón y cómo se relacionan entre sí.
- Ejemplo de Código: Presenta un ejemplo en un lenguaje popular, como Java o Python.

3

CLASIFICACIÓN DE LOS PATRONES

- Creacionales: Manejan la creación de objetos. Ejemplo: Singleton para garantizar que solo exista una instancia de una clase.
- Estructurales: Organizan la relación entre clases y objetos. Ejemplo: Adapter, que permite que dos interfaces incompatibles trabajen juntas.
- De Comportamiento: Facilitan la comunicación entre objetos. Ejemplo: Observar, que notifica cambios de estado entre objetos.

4

¿POR QUÉ SON IMPORTANTES LOS PATRONES DE DISEÑO?

- Eficiencia: Ayudan a resolver problemas de manera rápida y efectiva.
- Comunicación: Usar patrones crea un "lenguaje común" entre programadores. Por ejemplo, decir "usa un Singleton" comunica una idea clara sin explicación adicional.
- Reutilización de Soluciones: Los patrones permiten usar soluciones ya probadas, mejorando la calidad y evitando errores comunes.

5

HISTORIA DE LOS PATRONES DE DISEÑO

Origen: La idea de los patrones comenzó con Christopher Alexander en arquitectura y fue llevada a la programación en el libro Design Patterns: Elements of Reusable Object-Oriented Software, escrito por Erich Gamma, John Vlissides, Ralph Johnson, y Richard Helm, conocidos como la Gang of Four.

