



ANÁLISIS Y DISEÑO DE SOFTWARE

INTRODUCCIÓN A LOS PATRONES DE DISEÑO

INTRODUCCIÓN A LOS PATRONES DE DISEÑO



En el campo del desarrollo de software, los patrones de diseño representan soluciones generales, reutilizables y comprobadas para resolver problemas recurrentes relacionados con la estructuración del código y la interacción entre objetos. Estos patrones no son fragmentos de código específicos ni bibliotecas listas para usar, sino guías conceptuales que ofrecen buenas prácticas de diseño, orientado a objetos. Su aplicación correcta permite mejorar la calidad del software, promoviendo la reutilización, el bajo acoplamiento, la alta cohesión y la facilidad de mantenimiento (Sampedro Hernández, 2011).

Origen y propósito

El concepto de patrón de diseño fue popularizado por los autores Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, conocidos como la Banda de los Cuatro (Gang of Four, GoF), quienes en 1994 publicaron el influyente libro "Design Patterns: Elements of Reusable Object-Oriented Software". En él, se estableció una clasificación de 23 patrones organizados en tres grupos: creacionales, estructurales y de comportamiento. El propósito fundamental de estos patrones es brindar soluciones reutilizables a problemas frecuentes de arquitectura y codificación en aplicaciones orientadas a objetos (Piattini & Garzás, 2015).

Clasificación general de los patrones de diseño

1. **Patrones creacionales.** Se enfocan en los mecanismos para la creación de objetos, ayudando a abstraer el proceso de instanciación. Ejemplos: Singleton, Factory Method, Abstract Factory, Builder, Prototype.
2. **Patrones estructurales.** Describen cómo componer clases y objetos para formar estructuras más complejas, garantizando la flexibilidad del sistema. Ejemplos: Adapter, Composite, Decorator, Facade, Proxy.
3. **Patrones de comportamiento.** Se centran en cómo interactúan los objetos y cómo se comunican, definiendo patrones para la asignación de responsabilidades. Ejemplos: Observer, Strategy, Command, State, Iterator.

Importancia de los patrones de diseño

Implementar patrones de diseño permite que un sistema:

- Es más fácil de entender y mantener.
- Pueda adaptarse a cambios con menor esfuerzo.
- Promueva la reutilización de código.
- Facilite la comunicación entre desarrolladores mediante un vocabulario común (por ejemplo, decir que se usó un "Observer" implica una estructura bien entendida por otros programadores).