**Patrones de diseño**

Un patrón de diseño es una solución general y que puede reutilizarse en problemas que ocurren diariamente en el diseño de software. Es importante tener en cuenta que un patrón de diseño no es un diseño de software que ya está finalizado y listo para codificarse, más bien es una descripción o plantilla de cómo resolver un problema que puede ser utilizado en diferentes situaciones. Aquí es donde se presenta su importancia, dado que, estos patrones de diseño permiten agilizar el proceso de desarrollo de solución debido a que proveen un paradigma desarrollado y probado. Un diseño de software efectivo debe considerar detalles que tal vez no sean visibles hasta que se implemente la solución, es decir, debe anticiparse a los problemas y tratar de cubrir todos los que estén al alcance. Esto es muy importante, puesto que la reutilización de patrones de diseño ayuda a prevenir los pequeños detalles que provocarían problemas más grandes, además de ayudar a la legibilidad de código para programadores o analistas que estén familiarizados con patrones.

De manera general, un patrón de diseño posee 4 elementos esenciales:

1. El nombre del patrón se utiliza para describir el diseño del problema, su solución y sus consecuencias en una o dos palabras.

2. El problema describe cuando se debe aplicar el patrón. Explica el problema y su contexto.

3. La solución describe los elementos que forman el diseño, su relación, responsabilidades y colaboraciones

4. Las consecuencias son el resultado y las recompensas de haber aplicado el patrón.

Así mismo, dentro del catálogo de los patrones de diseño existen tres grandes grupos: los creacionales, los estructurales y los de comportamiento.

* Patrón de diseño creacional:

Este tipo de diseño abstrae el proceso de instanciación (creación de objetos), permitiendo con esto que el sistema sea independiente de cómo sus objetos son creados, se componen y se representan. Un patrón de diseño creacional de clases utiliza la herencia para variar la clase que crea una instancia, mientras que un patrón de diseño creacional de objetos delegará la instanciación a otro objeto.

Los patrones creacionales más conocidos son: Singleton, Factory Method, Abstract Factory, Builder y Prototype.

* Patrón de diseño estructural

A los patrones de diseño estructural les interesa como están compuestos clases y objetos para formar estructuras grandes. El patrón estructural de clases usa la herencia para componer la implementación de múltiples clases base. Esto es, cuando se mezclan varias clases base dentro de una (herencia múltiple), la clase resultante combina las propiedades de todas las clases base, generando con ello una estructura grande. Este patrón es muy útil cuando se quiere hacer que clases desarrolladas de manera independiente y en diferentes bibliotecas trabajen unidas.

El patrón estructural de objetos describe la manera en la que se componen los objetos para reaccionar ante una nueva funcionalidad. Los objetos poseen una composición flexible, esto es, tienen la habilidad de cambiar su forma en tiempo de ejecución, lo cual no es posible con una composición estática.

Los patrones estructurales más conocidos son: Composite, Proxy, Flyweight, Fecade, Bridge.

* Patrón de diseño de comportamiento

Los patrones de diseño de comportamiento están relacionados con los algoritmos y la asignación de responsabilidades entre objetos, es decir, no sólo describen los patrones de objetos o clases, también se encargan de los patrones de comunicación entre ellos. Este patrón se caracteriza por un complejo control de flujo, el cual es difícil manejar en tiempo de ejecución, por lo tanto, provoca que se cambie el enfoque lejos del control de flujo para concentrarse exclusivamente en la manera en la que los objetos se comunican.

Los patrones de comportamiento más conocidos son: Método Template, Interpreter, Mediator, Chain of responsability, Observer, Strategy, Command, State.

Referencia Bibliográfica:

Nakayama C., A., & Solano Gálvez , J. A. (s. f.). Guía práctica de estudio 13: Patrones de diseño. Recuperado de http://profesores.fi-b.unam.mx/annkym/LAB/poo\_p13.pdf