

El Impacto de Patrones de Diseño orientados a Objetos en el Desarrollo de Software

La programación orientada a objetos (POO) se puede tomar como una revolución en mejorar la calidad de los programas y códigos, además de realzar una nueva forma de programar, con nuevos conceptos, como el polimorfismo y la herencia de clases. Los patrones de diseño orientados a objetos se introdujeron a mitad de 1990 como una solución a los problemas comunes que tenían los patrones de diseño en ese momento. Un patrón de diseño funciona para dar solución a un problema de diseño que esté presente en el programa de un dominio en particular, sea telecomunicaciones, interfaces gráficas, bases de datos, etc.

Por lo general, se pueden reusar los mismos patrones de diseño en diversas ocasiones, si así la experiencia y el nivel arquitectural del programa lo permitan, pero claro, los patrones de diseño están presentes para dar solución a un problema presente, e incluyen también consecuencias al aplicar dicha solución. La solución se puede importar en diferentes lugares y con ello, acarrear las consecuencias (buenas o malas) de donde se hayan exportado. Por ello, se implementa el Diseño Orientado a Objetos, el cual está mejor estructurado y con más accesibilidad para su exportación.

Al implementar el Diseño Orientado a Objetos, se debe elegir una clase en donde partir, para poder llegar al final del proyecto o trabajo de la forma más eficaz posible. Existen 3 tipos de clases básicas en lo que patrones de diseño se refieren. Está la Estructural, la cual identifica la manera más sencilla de encontrar relaciones entre los objetos presentes; la Creacional, en donde se basa, no solo en guardar cuáles clases concretas se usarán por el sistema, sino que también oculta cómo las instancias de dichas clases son creadas y combinadas; y por último la Conductual, que identifica la comunicación más común entre objetos y les asigna responsabilidades entre ellos. Con tantos patrones de diseños para escoger, puede ser difícil elegir el que más se adapte a nuestro estilo o proyecto, por lo que se recomienda investigar sobre los posibles candidatos a ser el patrón de diseño adecuado, para ganar suficiente información sobre él (¿cómo se usa?, ¿se logra entender la lógica?, ¿se puede implementar en mi programa?) y hacer una decisión final, que decidirá la ruta que el proyecto tome, llegando con consecuencias buenas o malas.

Al elegir el patrón de diseño que mejor se adapta al proyecto en el cual se está trabajando, se debe elegir el cómo aplicarlo. Con el patrón de diseño en cuestión, se debe investigar sobre su aplicación y consecuencias de usarse, así como el entendimiento de todo lo que lo conforma (clases, atributos, objetos, etc.) y del código donde se implementa dicho patrón de diseño. Con todo esto en cuenta, se puede comenzar a diseñar, definir y crear las clases e interfaces necesarias para crear el patrón de diseño mejor adaptado y concluir el proyecto.

Existen diversas ventajas y desventajas de los patrones de diseños, por ejemplo, como ventajas están que pueden reducir considerablemente el tiempo de desarrollo del proyecto. También poseen flexibilidad y extensibilidad, la cual es importante para poder comunicarse con diversos equipos de desarrollo más fácilmente. Se pueden reusar fácilmente, eliminando las alternativas que no tienen reutilización. Existen estas y más ventajas, pero también tienen sus desventajas, como las siguientes: El uso indebido de patrones de diseño, indicando que se puede dar el error de implementar un patrón de diseño donde no se necesita. Y bien, el uso de patrones de diseño no es necesariamente para aumentar la calidad del proyecto, puesto que se pueden implementar mal o innecesariamente, causando que un proyecto entendible a simple vista sea más complicado de comprender.

Como se pudo observar antes, los patrones de diseño se pueden implementar en proyecto donde se pueden usar y que aporten ventajas sobre todo en él. Aunque como hay ventajas, también existen desventajas con ellos si no se aplican como debe de ser. Los patrones de diseño pueden ser complicados de entender de primeras, pero con suficiente investigación y experiencia se pueden lograr usar correctamente cada vez que se necesite.