**PATRONES DE DISEÑO**

Los patrones de diseño en Java son soluciones reutilizables y probadas a problemas comunes que se encuentran en el desarrollo de software. Estos patrones son una forma de capturar prácticas de diseño exitosas para que puedan ser aplicadas en diferentes contextos. Los patrones de diseño ayudan a escribir código más flexible, reutilizable y mantenible.

Los patrones de diseño en Java son, en definitiva, soluciones a problemas recurrentes y que se ha documentado que funcionan y los resuelven.

**Clasificación de los Patrones de Diseño:**

* Patrones Creacionales:

Los patrones de creación se utilizan para crear objetos para una clase adecuada que sirva como solución a un problema.

Algunos Patrones Creacionales son:

* Abstract Factory
* Singleton
* Patrones Estructurales:

Los patrones estructurales forman estructuras más grandes a partir de elementos únicos, generalmente de diferentes clases.

Algunos Patrones Estructurales son:

* Adapter
* Composite
* Patrones de Comportamiento:

Los patrones de comportamiento describen interacciones entre objetos y se centran en cómo los objetos se comunican entre sí. Pueden reducir los diagramas de flujo complejos a simples interconexiones entre objetos de varias clases

Algunos patrones de comportamiento son:

* Observer
* Command