

# PATRONES DE DISEÑO EN PROGRAMACIÓN

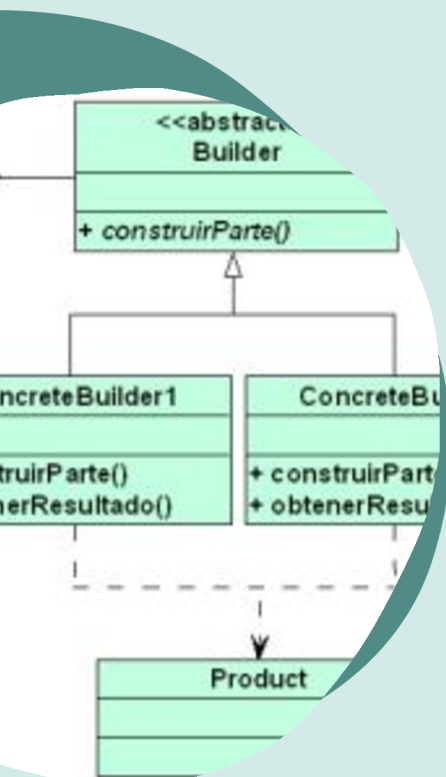


## ¿Qué son los Patrones de Diseño?

- Los patrones de diseño son soluciones probadas para problemas comunes en el desarrollo de software.
  - Ayudan a mejorar la calidad, flexibilidad y mantenibilidad del código.
- Los cerros se expresan en cada pieza original, ponchos y tejidos.

## Clasificación de Patrones de Diseño

- Patrones Creacionales: Facilitan la creación de objetos, promoviendo flexibilidad y reutilización.
- Patrones Estructurales: Se enfocan en la composición de clases y objetos para formar estructuras más grandes.
- Patrones de Comportamiento: Mejoran la comunicación entre objetos y la asignación de responsabilidades.



## Ejemplos de Patrones Comunes

Patrones Creacionales:

- Singleton: Asegura una única instancia de una clase.
- Factory Method: Crea objetos sin especificar el tipo exacto.
- Builder: Permite construir objetos complejos paso a paso.

Patrones Estructurales:

- Adapter: Permite que clases con interfaces incompatibles trabajen juntas.
- Decorator: Agrega funcionalidad a un objeto de forma dinámica.
- Facade: Proporciona una interfaz simplificada para un sistema complejo.

## Patrones de Comportamiento:

- Observer: Define una dependencia uno-a-muchos para que un cambio en un objeto notifique a otros.
- Strategy: Permite seleccionar un algoritmo en tiempo de ejecución.
- Command: Encapsula una solicitud como un objeto, permitiendo parametrizar otros objetos con operaciones.



## ¿Por Qué Usar Patrones de Diseño?

- Reutilización de Código: Soluciones que se pueden aplicar en diferentes proyectos.
- Facilidad de Mantenimiento: Hace que el código sea más comprensible y fácil de modificar.
- Buenas Prácticas: Fomentan el uso de buenas prácticas de desarrollo.