

## **Aplicando Patrones en la solución de problemas en POO**

Cristian Esteban Ruiz Parra

Ingeniería de Software, Universidad de Cundinamarca

Programación II

Ebelin Rodriguez Lombo

21 de abril de 2025

Patrones de Diseño de Software en la POO

Tipos de Patrones de Diseño

Clasificación General

Los patrones de diseño se dividen en tres categorías principales:

Patrones Creacionales: Se ocupan de la creación de objetos.

Patrones Estructurales: Se centran en la composición de clases y objetos.

Patrones de Comportamiento: Se enfocan en la interacción entre objetos.

Ejemplos de Patrones

Creacionales: Singleton, Factory Method, Abstract Factory.

Estructurales: Adapter, Composite, Proxy.

Comportamiento: Observer, Strategy, Command.

Patrones de Diseño de Software

Aplicación en Programación Orientada a Objetos (POO)

Los patrones de diseño son soluciones reutilizables a problemas comunes en el desarrollo de software.

En POO, estos patrones permiten mejorar la organización, la mantenibilidad y la escalabilidad del código.

Ejemplo de Patrón de Diseño: Singleton

Definición y Propósito

El patrón Singleton asegura que una clase tenga una única instancia y proporciona un punto de acceso global a esa instancia.

Es útil para gestionar recursos compartidos o configuraciones globales.

Características Principales

Instancia Única: Solo se crea una instancia de la clase.

Acceso Global: La instancia se puede acceder desde cualquier parte del código.

Control de la Creación: Previene la creación de múltiples instancias a través de un constructor privado.

Elementos Identificadores

Método Estático: Proporciona acceso a la instancia.

Constructor Privado: Evita la instanciación externa.

Inicialización Tardía: La instancia se crea solo cuando se necesita.

Ejemplo de Implementación

En un sistema de configuración, se puede usar el patrón Singleton para garantizar que la configuración sea consistente en toda la aplicación.

Código de ejemplo en Java: