**Informe Método Prototype**

**Joshua Abril**

**Patrones de Diseño**

**Luis Alberto Reyes**

**Uninpahu**

**2025**

**¿Qué es?:** El patrón **Prototype** es un patrón de diseño creacional que permite copiar objetos existentes sin que el código cliente dependa de sus clases. En lugar de crear una nueva instancia con new, se clona un objeto ya existente, lo que puede ser más eficiente cuando la creación es costosa en términos de tiempo o recursos.

**Funcionalidad:** El método Prototype se basa en la implementación de una interfaz de clonación, generalmente con un método como clone(). Cada clase que desea ser clonable implementa este método para devolver una copia de sí misma. Esto permite crear nuevos objetos duplicando prototipos ya configurados, ideal cuando se desea mantener la configuración o estructura de un objeto base.

**Ejemplo breve:** Una clase Documento con múltiples configuraciones. Si deseas crear varios documentos similares, puedes simplemente clonar uno ya configurado en lugar de crear y configurar cada uno desde cero.

**Código de Ejemplo:**

interface Prototipo {

Prototipo clonar();

void mostrar();

}

class Documento implements Prototipo {

private String tipo;

public Documento(String tipo) {

this.tipo = tipo;

}

public Prototipo clonar() {

return new Documento(this.tipo);

}

public void mostrar() {

System.out.println("Tipo de documento: " + tipo);

}

}

public class Main {

public static void main(String[] args) {

Documento original = new Documento("Informe");

Documento copia = (Documento) original.clonar();

original.mostrar();

copia.mostrar();

}

}

**Explicación del Ejemplo del Repositorio**

El repositorio implementa el patrón Prototype para crear copias de objetos de forma eficiente. En lugar de instanciar nuevos objetos desde cero, se parte de un objeto base y se clona. Esto es especialmente útil cuando los objetos tienen estructuras internas complejas o inicialización costosa. La clase prototipo define el método clonar() y cada subclase lo implementa según su lógica de copia. En la clase principal (Main), se crean clones de distintos objetos, los cuales mantienen el estado del objeto base. Esto promueve la reutilización, reduce el acoplamiento y mejora la performance en ciertos escenarios.