

# Trabajo Práctico – Patrones Creacionales

## Objetivo

- Aplicar los cinco patrones creacionales en Java.
- Comprender diferencias entre creación de objetos y clonación.
- Ver cómo cada patrón facilita la flexibilidad, escalabilidad y mantenimiento del código.

## Contexto del ejercicio

Vas a desarrollar un **mini sistema de biblioteca online**, con funcionalidades básicas:

1. Gestión de libros y usuarios.
2. Creación de objetos complejos de forma controlada.
3. Capacidad de clonar objetos cuando sea necesario.
4. Selección de familias de objetos según el contexto (ej: diferentes tipos de UI o métodos de envío).

## Parte 1 – Singleton

**Objetivo:** Garantizar que la **base de datos de libros** tenga **una sola instancia** en toda la aplicación.

### Tareas:

1. Crear la clase Database como **Singleton**.
2. Implementar métodos para agregar y listar libros.
3. Probar que múltiples llamadas devuelvan **la misma instancia**.

**NOTA:** No es necesario implementar una conexión a una base de datos real, puede hacer uso de por ejemplo una variable `List<Libro> libros` para emular la base

## Parte 2 – Factory Method

**Objetivo:** Crear diferentes **tipos de libros** sin acoplar el cliente a la clase concreta.

### Tareas:

1. Crear la interfaz Libro y las clases concretas LibroFisico y LibroDigital.
2. Crear el método fábrica en LogisticaLibro que devuelva la instancia correcta según un parámetro.
3. Probar la creación de varios libros y mostrar su tipo en consola.

## Parte 3 – Abstract Factory

**Objetivo:** Crear **familias de objetos relacionados** (ej: UI o métodos de envío) según el tipo de usuario.

### Tareas:

1. Crear interfaces InterfazUI y MetodoEnvio.
2. Implementar AdminUI/UsuarioUI y EnvioNormal/EnvioExpress.
3. Crear AbstractFactory y fábricas concretas (AdminFactory, UsuarioFactory).
4. Usar la fábrica para crear objetos correctos según el tipo de usuario y mostrar información en consola.

## Parte 4 – Builder

**Objetivo:** Construir objetos **complejos** de manera clara y flexible.

### Tareas:

1. Crear la clase Usuario con atributos opcionales: nombre, email, dirección, teléfono, fecha de nacimiento.
2. Implementar la clase Usuario.Builder para construir instancias.
3. Crear al menos 2 usuarios con distintos atributos y mostrar la información.

## Parte 5 – Prototype

**Objetivo:** Clonar objetos existentes para crear **nuevos objetos similares**.

### Tareas:

1. Crear la clase Prestamo con atributos: libro, usuario, fechaInicio, fechaFin.
2. Implementar clone() (shallow o deep).
3. Crear un prestamo prototipo y clonar al menos dos veces, modificando solo algunos atributos.
4. Mostrar en consola que los clones son independientes del original.

### Entregables

1. Proyecto completo en **IntelliJ IDEA**.
2. Archivos .java organizados por paquetes.

LA estructura del proyecto debe ser similar o igual a:

```
BibliotecaCreacional/  
├─ src/  
│   └─ main/  
│       └─ java/  
│           ├── singleton/  
│           │   └─ Database.java  
│           ├── factory/  
│           │   ├── Libro.java  
│           │   ├── LibroFisico.java  
│           │   ├── LibroDigital.java  
│           │   └─ LogisticaLibro.java  
│           ├── abstractfactory/  
│           │   ├── AbstractFactory.java  
│           │   ├── AdminFactory.java  
│           │   ├── UsuarioFactory.java  
│           │   ├── InterfazUI.java  
│           │   ├── MetodoEnvio.java  
│           │   ├── AdminUI.java  
│           │   ├── UsuarioUI.java  
│           │   ├── EnvioNormal.java  
│           │   └─ EnvioExpress.java  
│           ├── builder/  
│           │   └─ Usuario.java  
│           ├── prototype/  
│           │   └─ Prestamo.java  
│           └─ Main.java
```