# Diseño de Software: Patrón Factory

## Resumen General

El patrón **Abstract Factory** (Fábrica Abstracta) es un patrón de diseño creacional que permite la creación de familias de objetos relacionados sin especificar sus clases concretas. A diferencia del **Factory Method**, que se centra en un único tipo de objeto, el **Abstract Factory** gestiona múltiples objetos interdependientes, asegurando su compatibilidad y coherencia.

**Propósito Principal**

Proporcionar una interfaz unificada para crear conjuntos de objetos que pertenecen a una misma familia, evitando que el cliente dependa de implementaciones específicas.

**Ejemplo Práctico: Conexiones a Bases de Datos**

Supongamos un sistema que requiere conexiones a diferentes motores de bases de datos (MySQL, Oracle, SQL Server). Cada motor tiene objetos asociados como *Conexion*, *Comando* y *Resultado*.

**Implementación Clave**

1. **Interfaz Abstracta (**FabricaAbstracta**):**  
   Define métodos para crear los objetos de la familia (ej: crearConexion(), crearComando()).
2. **Fábricas Concretas (**ConexionMySQLFabrica**,**ConexionOracleFabrica**):**  
   Implementan la interfaz y producen los objetos específicos para cada motor.

java

Copy

// Ejemplo en Java

FabricaAbstracta fabrica = new ConexionMySQLFabrica();

Conexion conexion = fabrica.crearConexion(); // Devuelve una conexión MySQL

Comando comando = fabrica.crearComando(); // Devuelve un comando MySQL

**Ventajas del Patrón**

* **Consistencia:** Garantiza que los objetos creados sean compatibles (ej: una conexión MySQL con un comando MySQL).
* **Flexibilidad:** Facilita el cambio de familias completas (ej: migrar de MySQL a Oracle).
* **Desacoplamiento:** El cliente solo interactúa con interfaces abstractas.

**Diagramas Explicativos**

1. **Diagrama de Clases:**
   * Muestra la jerarquía entre la fábrica abstracta, las fábricas concretas y los productos.
2. **Diagrama de Secuencia:**
   * Ilustra cómo el cliente obtiene objetos de una familia sin conocer sus implementaciones.
3. **Diagrama de Contexto:**
   * Explica el uso del patrón en un escenario real, como un sistema multi-motor de bases de datos.

**Conclusión**

El **Abstract Factory** es ideal para sistemas que manejan múltiples variantes de objetos interdependientes. Su capacidad para abstraer la creación de familias enteras lo convierte en una herramienta poderosa para mantener código limpio, escalable y fácil de modificar.