Patron factory

**Introducción al Patrón Factory**

El patrón **Factory** es un enfoque de diseño que centraliza la creación de objetos en una clase especializada, conocida como "fábrica". Este mecanismo permite generar instancias sin revelar los detalles de construcción al cliente, lo que promueve un código más limpio, desacoplado y escalable.

**Ejemplo Práctico**

Imaginemos un sistema de notificaciones donde se requieren diferentes tipos de mensajes, como correos electrónicos, SMS o notificaciones push. En lugar de que el cliente cree estos objetos directamente, una clase **FabricaNotificaciones** se encargaría de generarlos mediante un método único, como:

java

Copy

Notificacion notificacion = FabricaNotificaciones.crearNotificacion("email");

Esta abstracción simplifica el código y facilita la incorporación de nuevos tipos de notificaciones en el futuro.

**Ventajas del Patrón Factory**

* **Reducción del acoplamiento:** El cliente no necesita conocer los detalles de creación de los objetos.
* **Escalabilidad:** Añadir nuevos tipos de objetos es sencillo y no afecta al código existente.
* **Mantenibilidad:** La lógica de creación está centralizada, lo que facilita las modificaciones.

**Diagramas Ilustrativos**

A continuación, se presentan diagramas que ejemplifican la estructura y el flujo del patrón Factory:

1. **Diagrama de Clases:** Muestra la relación entre la fábrica y los productos.
2. **Diagrama de Secuencia:** Ilustra el proceso de creación de un objeto mediante la fábrica.
3. **Diagrama de Flujo:** Explica las decisiones que toma la fábrica para instanciar objetos.





