

DISSO 8 – PATRONES DE CREACIÓN

PATRONES DE CREACIÓN

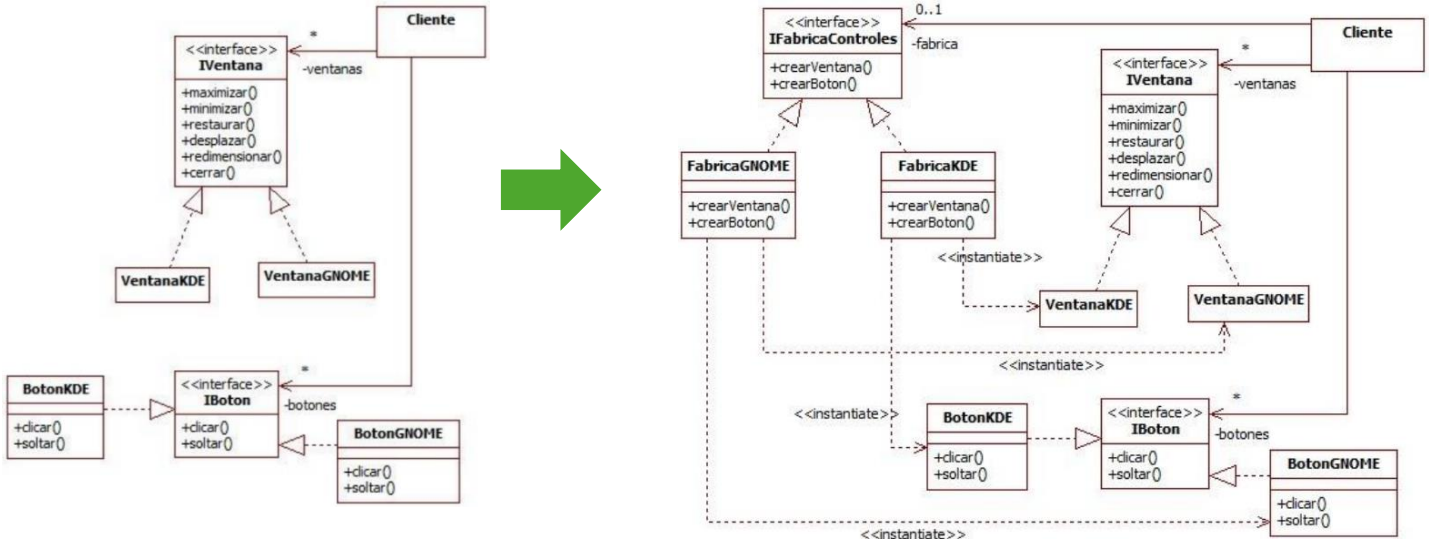
- Los PATRONES DE CREACIÓN **abstraen** el proceso de **creación de objetos**, acotando el **conocimiento** sobre las clases concretas usadas.
- Ocultan cómo se **crean y enlazan** las instancias de las clases, de manera que el **resto del sistema sólo conoce los objetos a través de sus interfaces**.
- ▶ Hacen posible **programar para interfaces** y no para implementaciones.

ABSTRACT FACTORY

- La FÁBRICA ABSTRACTA aporta una **interfaz** para crear **familias de objetos** ocultando clases concretas.

Se usa para:

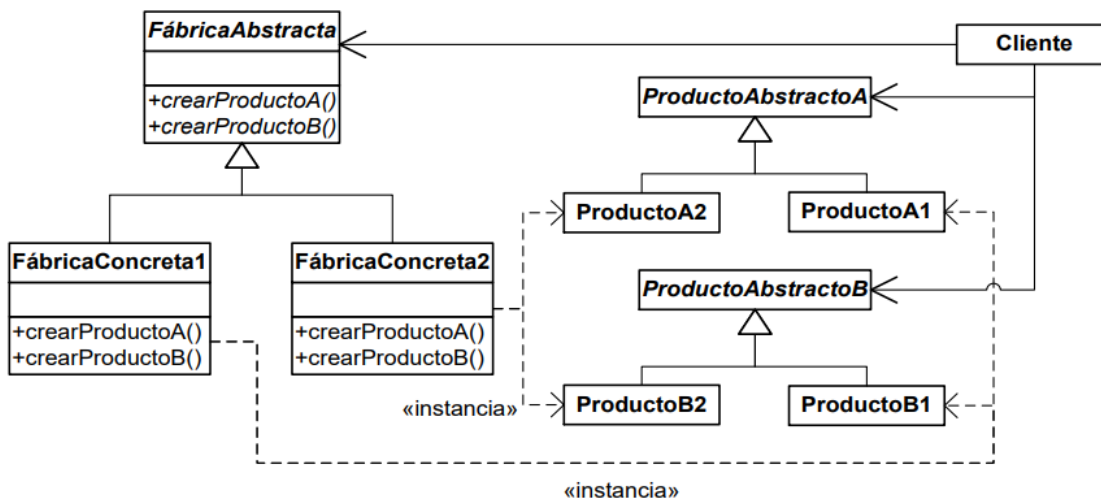
- **Configurar** un sistema escogiendo una sola **familia de productos** de entre varias.
- Si cada **familia** está diseñada para el **uso conjunto de sus productos**.
- Para proporcionar **bibliotecas de productos** de los que sólo se revelan sus interfaces.



PARTICIPANTES

- FÁBRICA ABSTRACTA → declara una **interfaz** con operaciones para **creación de productos**.
- FÁBRICA CONCRETA → **implementa** operaciones para **crear productos concretos**.
- PRODUCTO ABSTRACTO → declara una **interfaz** para un **tipo de producto**.
- PRODUCTO CONCRETO → define un **producto** a ser creado por la fábrica correspondiente (**implementa Producto Abstracto**).
- CLIENTE → usa objetos a través de interfaces.

ESTRUCTURA



IMPLEMENTACIÓN

- En el folio.

VENTAJAS E INCONVENIENTES

- ✓ **Desacoplamiento entre clientes y productos** → los clientes manipulan las instancias a través de las interfaces.
- ✓ **Facilidad de cambio de familias.**
- ✓ **Consistencia** → las aplicaciones sólo generan productos de la misma familia.
- ✗ **Introducción problemática de nuevos productos.**

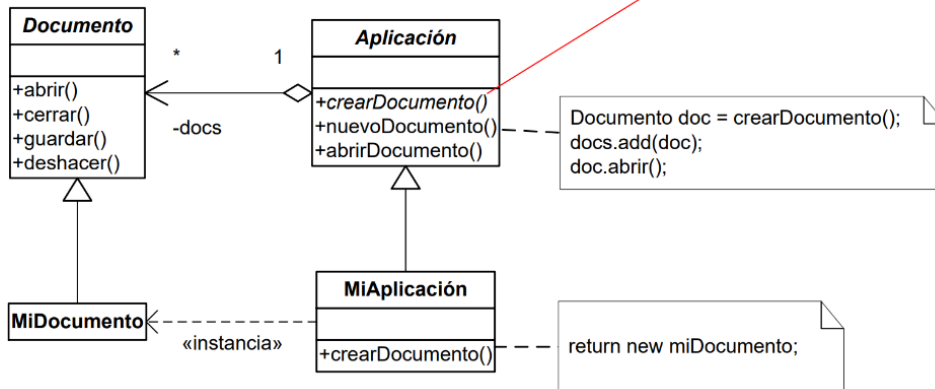
FACTORY METHOD

- El MÉTODO DE FABRICACIÓN define una **interfaz** para **crear objetos** cediendo a las **subclases** la decisión sobre **qué clase instanciar**.

Se usa cuando:

- Una clase no puede prever la clase de objetos que debe crear.

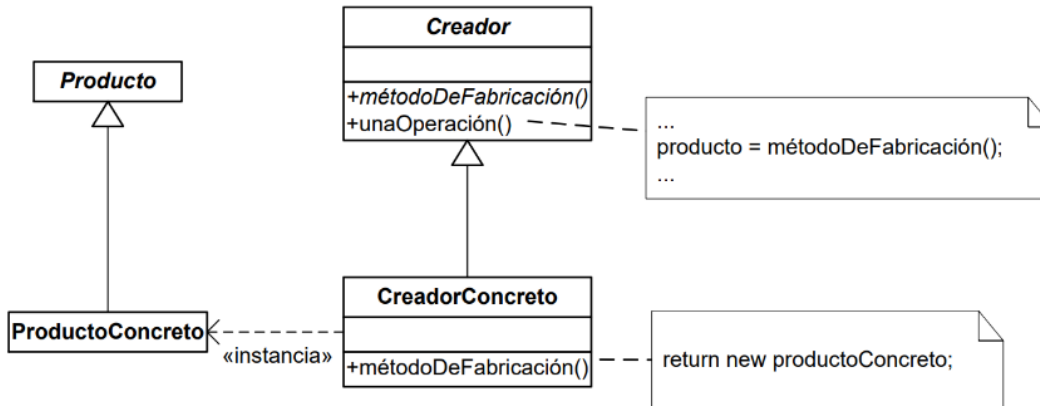
Método de fabricación



PARTICIPANTES

- PRODUCTO → define la **interfaz** de los objetos creados por el **método de fabricación**.
- PRODUCTO CONCRETO → **implementa** la interfaz **Producto**.
- CREADOR → **declara** (y puede implementar) el **método de fabricación**, el cual devuelve un objeto **Producto**.
- CREADOR CONCRETO → **redefine** el **método de fabricación** para devolver una instancia de **Producto Concreto**.

ESTRUCTURA



IMPLEMENTACIÓN

- En el folio.

VENTAJAS E INCONVENIENTES

- ✓ Los **clientes** sólo tratan con la **interfaz Producto**.
- ✓ Dota a **subclases** de un enganche para proveer **versión extendida** de un **objeto**.
- ✓ Puede conectar **jerarquías paralelas**.
- ✗ Los clientes pueden tener que **heredar** de **Creador** simplemente para **crear** un determinado **Producto Concreto**.

COMPARACIÓN ENTRE PATRONES

- **Método de fabricación** → heredar de la clase que crea los objetos.
 - ↳ Hace más adaptable el diseño a cambio de un poco de complejidad.
- **Fábrica abstracta** → definir un objeto responsable de conocer la clase de los productos.
 - ↳ Da lugar a un diseño aún más flexible y complejo.