

*Patrones de Diseño:
Patrones de Creación.
Tema 3-2:
Abstract Factory*

Descripción del patrón

- **Nombre:**
 - Fábrica Abstracta
 - Conocido como: Kit, Toolkit
- **Propiedades:**
 - Tipo: creación
 - Nivel: objeto, componente
- **Objetivo o Propósito:**
 - Proporciona una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.

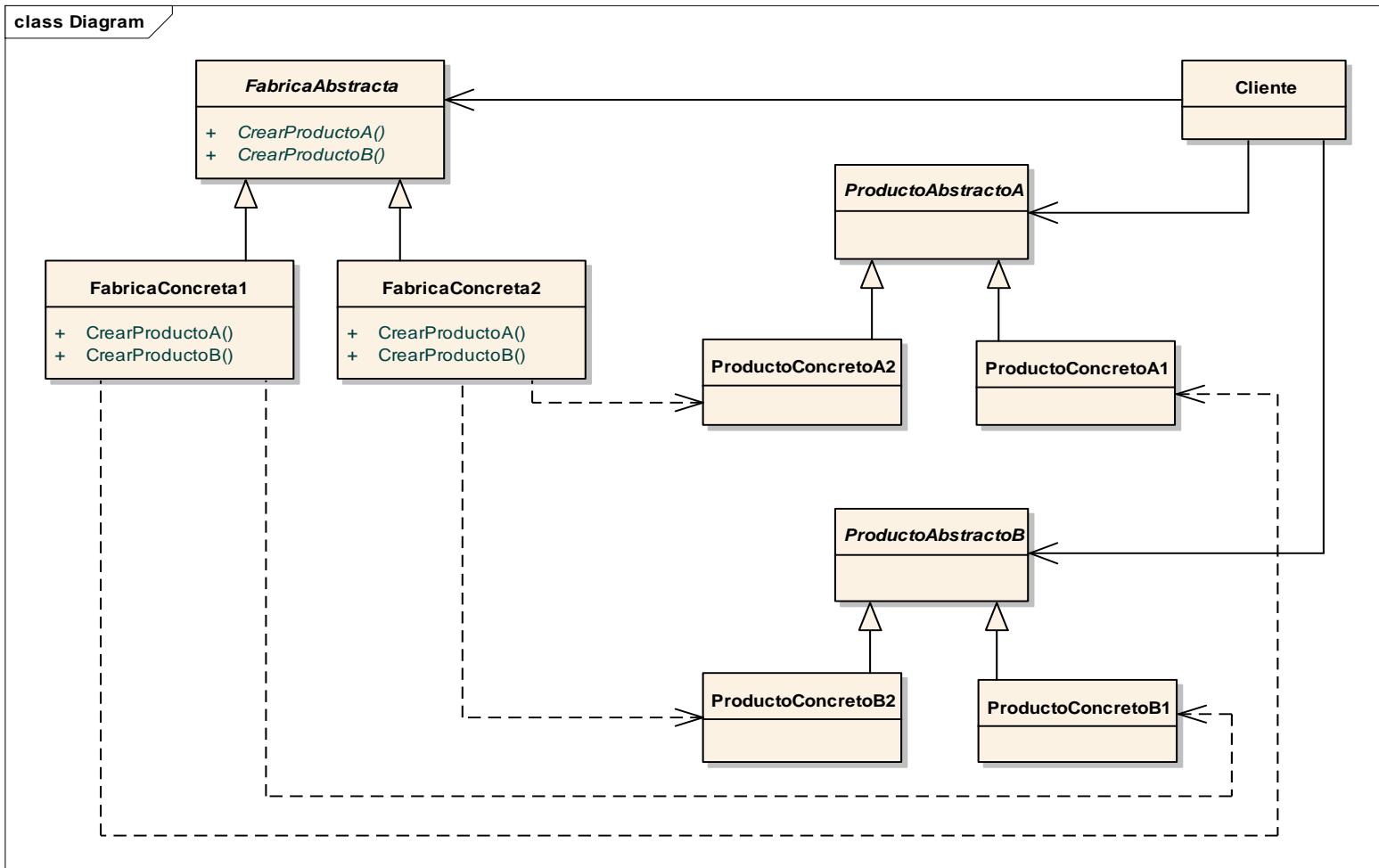


Aplicabilidad

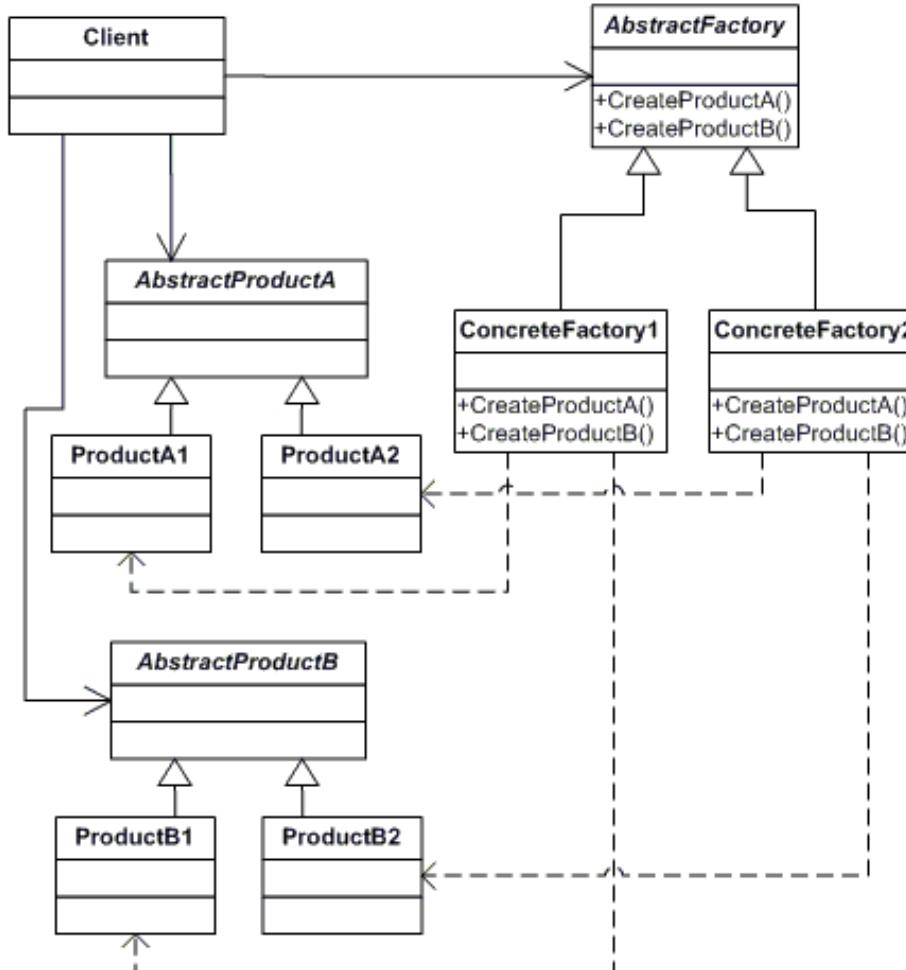
- Use el patrón Abstract Factory cuando:
 - El cliente debe ser independiente del proceso de creación de los productos.
 - La aplicación debe configurarse con una o más familias de productos.
 - Es necesario crear los objetos como un conjunto, de forma que sean compatibles.
 - Se desea proporcionar una colección de clases y únicamente quiere revelar sus interfaces y sus relaciones, no sus implementaciones.
- Ejemplo: Sistema de gestión de ventanas o sistema de ficheros para diversos sistemas operativos.



Estructura



Estructura

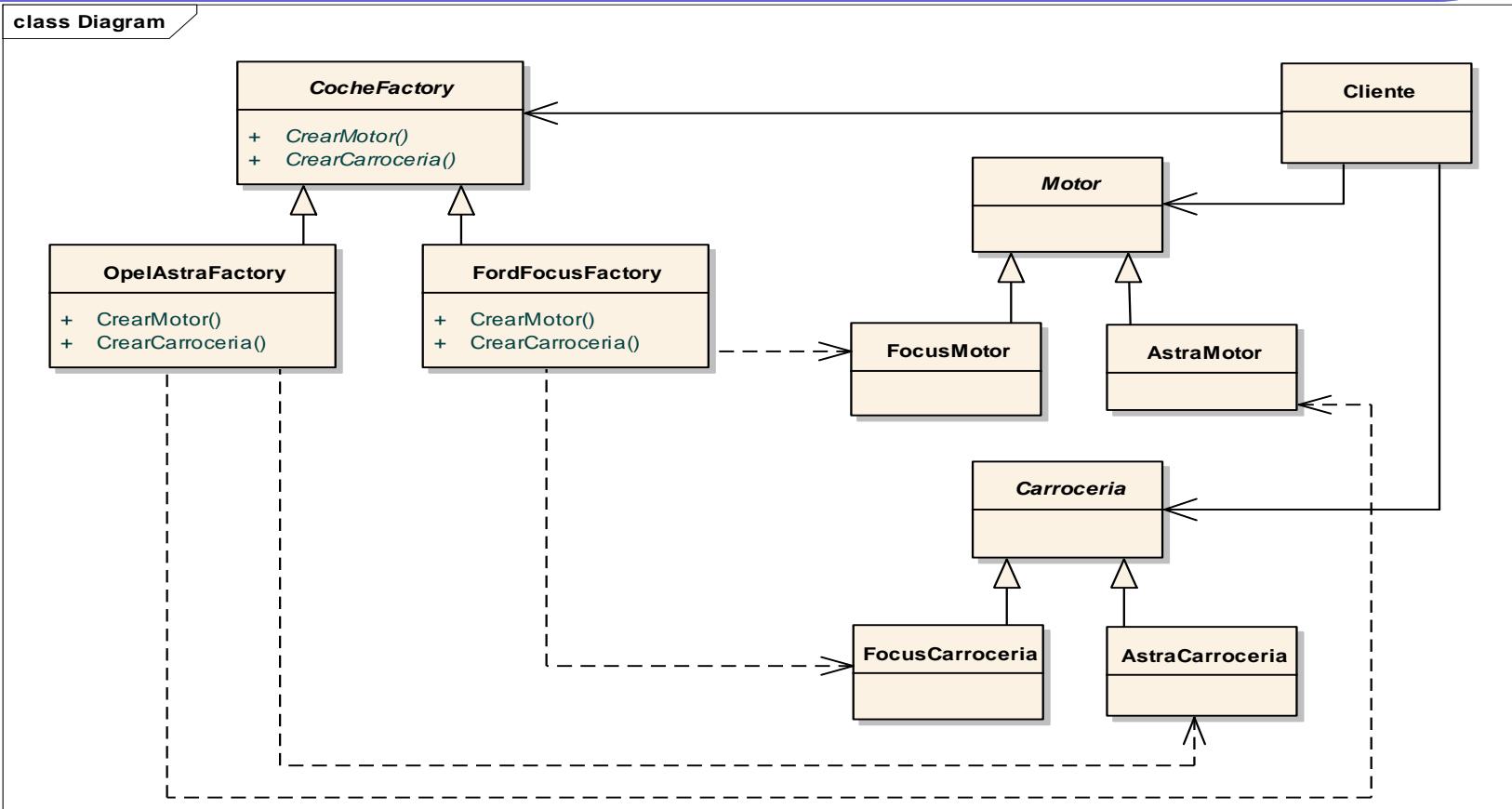


Estructura. Participantes

- **FabricaAbstracta:** Declara una interfaz para operaciones que crean objetos de tipo producto abstracto.
- **FabricaConcreta:** Implementa las operaciones para crear objetos de tipo producto concreto.
- **ProductoAbstracto:** Declara una interfaz para un tipo de objeto producto.
- **ProductoConcreto:** Define un objeto producto para que sea creado por la fábrica correspondiente. Implementa la interfaz ProductoAbstracto.
- **Cliente:** Solo usa interfaces declaradas por las clases FabricaAbstracta y ProductoAbstracto.



Estructura Ejemplo



Estructura Ejemplo

- Identificamos a continuación los elementos del patrón Abstract Factory:
 - FabricaAbstracta: CocheFactory.
 - FabricaConcreta: OpelAstraFactory, FordFocusFactory.
 - ProductoAbstracto: Motor, Carrocería.
 - ProductoConcreto: FocusMotor, AstraMotor, FocusCarroceria, AstraCarroceria.
 - Cliente: Cliente.



Consecuencias

- 1. Aísla las clases concretas.** Ayuda a controlar las clases de objetos que crea una aplicación. Aísla a los clientes de las clases de implementación. Los clientes manipulan las instancias a través de sus interfaces abstractas.
- 2. Facilita el intercambio de familias de productos.** Como se crea una familia completa de productos, toda la familia de productos cambia de una vez.
- 3. Promueve la consistencia entre productos.** Facilita que se trabaje con una sola familia de productos.
- 4. Dificulta la incorporación de nuevos tipos de productos.** Debido a que la interfaz FabricaAbstracta fija el conjunto de productos que se pueden crear.



Patrones relacionados

- **Factory Method y Prototype:** Las clases FabricaAbstracta suelen implementarse con métodos de fabricación pero también se pueden implementar usando prototipos.
- **Singleton:** Una fábrica concreta suele ser un Singleton.
- **Builder.** El patrón Abstract Factory se preocupa de lo que se va a hacer, y el Builder, de cómo se va a hacer. Abstract Factory es similar al patrón Builder en que ambos pueden construir objetos complejos. La principal diferencia es que el patrón Builder se centra en construir un objeto complejo paso a paso. Abstract Factory hace énfasis en familias de objetos Producto, tanto simples como complejas.



Conclusiones

- **Ocultación de las clases de implementación.** Esto también permite ofrecer clases sin mostrar su implementación, pudiéndonos reservar la posibilidad de cambiar la implementación en el futuro.
- Facilidad de **sustitución** de conjuntos de clases por otras equivalentes.
- **Consistencia** entre productos, al desarrollarse las aplicaciones con un mismo conjunto homogéneo de clases, dichas aplicaciones son más consistentes entre ellas.
- **Soportar nuevos tipos de productos es difícil.** Extender los Abstract Factory para producir nuevos tipos de Productos no es fácil. Soportar nuevos tipos de productos requiere extender la interface factory, que implica cambiar la clase Abstract Factory y todas sus subclases.

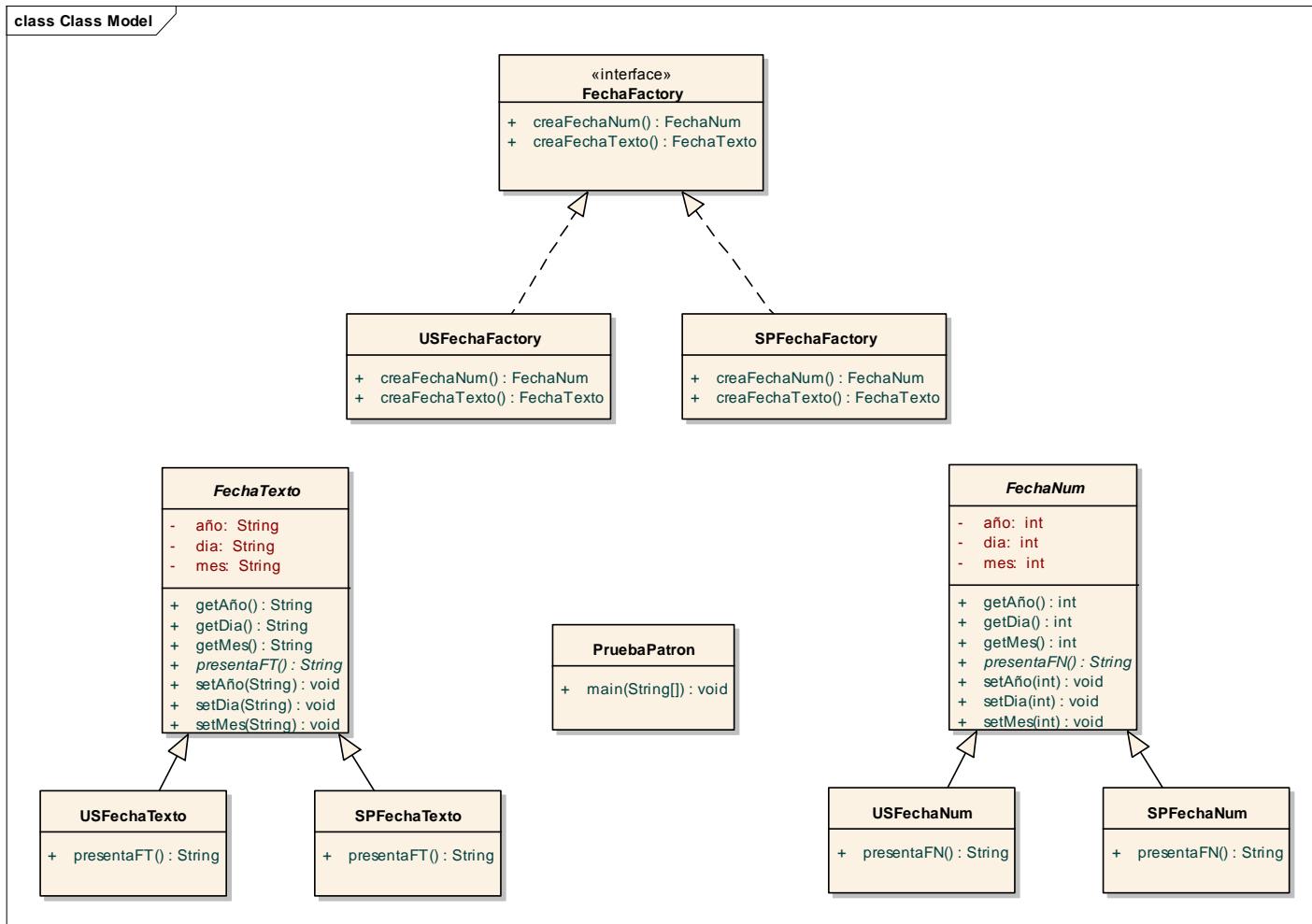


Código de ejemplo

Factoría de fechas



Código de ejemplo



Código de ejemplo

- Identificamos a continuación los elementos del patrón Abstract Factory:
 - FabricaAbstracta: FechaFactory.
 - FabricaConcreta: USFechaFactory, SPFechaFactory.
 - ProductoAbstracto: FechaTexto, FechaNum.
 - ProductoConcreto: USFechaTexto , SPFechaTexto, USFechaNum, SPFechaNum.
 - Cliente: PruebaPatron.

