



PATRÓN ABSTRACT FACTORY

DISEÑO DE SISTEMAS – UTNFRBA – 2020
Ing. Paula Zanetti

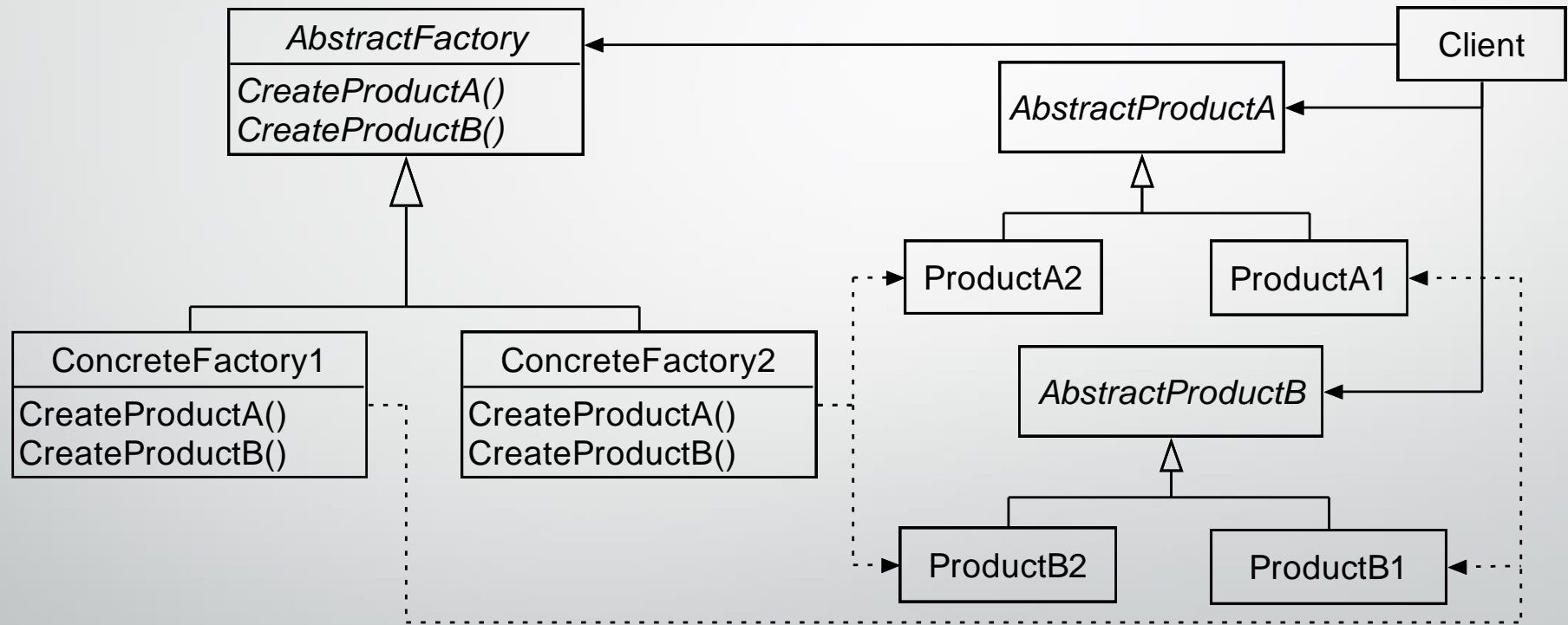
PROPÓSITO

“Proporciona una interfaz para crear familias de objetos relacionados ó que dependen entre sí, sin especificar sus clases concretas”

APLICABILIDAD

- Un sistema debe ser independiente de cómo se crean, componen y representan sus productos
- Un sistema debe ser configurado con una familia de productos de entre varias
- Una familia de objetos relacionados está diseñada para ser usada conjuntamente, y es necesario hacer cumplir esta restricción
- Quiere proporcionar una biblioteca de clases, y sólo quiere revelar sus interfaces, no sus implementaciones

ESTRUCTURA



PARTICIPANTES

Las clases y / u objetos que participan en este patrón

AbstractFactory

declara una interfaz para las operaciones que crean productos abstractos

ConcreteFactory

implementa las operaciones para crear objetos de productos concretos

AbstractProduct

declara una interfaz para un tipo de producto objeto

Producto

define un objeto producto a ser creado por la fábrica correspondiente

implementa la interfaz AbstractProduct

Client

utiliza las interfaces declaradas por clases AbstractFactory y AbstractProduct

VENTAJAS

- **Aísla las clases concretas para el cliente**
- **Facilita el intercambio entre familias de clase**
- **Promueve la consistencia entre productos**
- **Delega la creación de objetos producto en su subclase FabricaConcreta**

DESVENTAJAS

- No es sencillo agregar elementos a la factoría, ampliar las fábricas abstractas para producir nuevos tipos de productos no es fácil
- Requiere modificar la interface
- Requiere modificar las subclases

PATRONES RELACIONADOS

- Las clases de `FabricaAbstracta` suelen implementarse con métodos de fabricación (Patrón Factory Method) pero también se puede implementar usando prototipos (Patrón Prototype)

Ejemplo

