

Universidad politécnica salesiana

Nombre: Andrés Alvarado

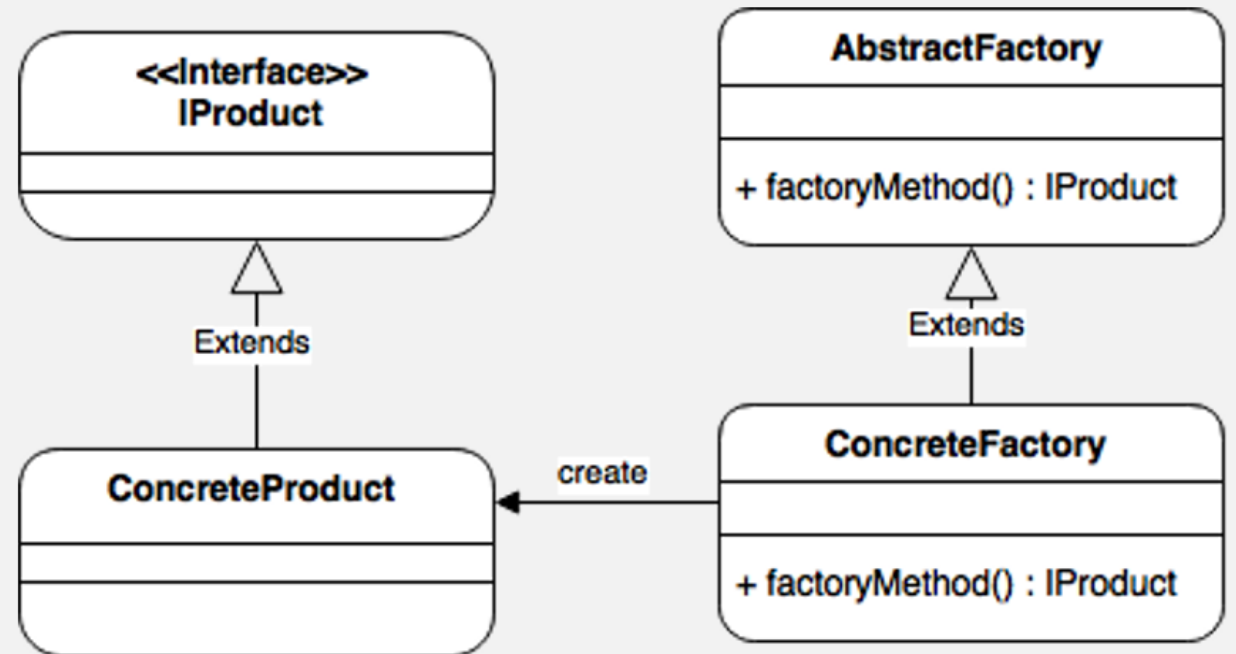
Materia: Programación aplicada

Tema: Factory Method

Factory Method

- Es un patrón de diseño creacional que proporciona una interfaz para crear objetos en una superclase, mientras permite a las subclasses alterar el tipo de objetos que se crearán.

Factory Method – Class diagram



Aplicabilidad

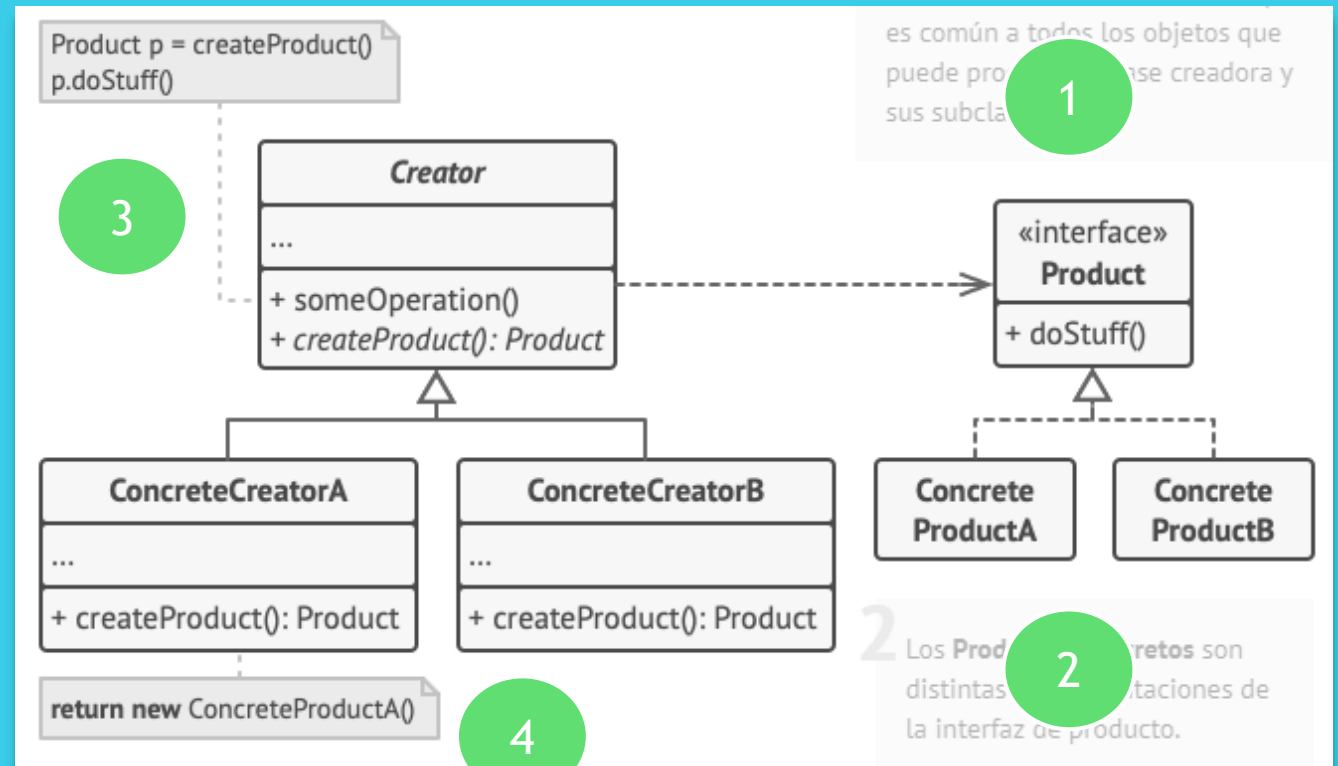
Utiliza el Método Fábrica cuando no conozcas de antemano las dependencias y los tipos exactos de los objetos con los que deba funcionar tu código.

Utiliza el Factory Method cuando quieras ofrecer a los usuarios de tu biblioteca o framework, una forma de extender sus componentes internos

Utiliza el Factory Method cuando quieras ahorrar recursos del sistema mediante la reutilización de objetos existentes en lugar de reconstruirlos cada vez.

Estructura

- 1.El **Producto** declara la interfaz, que es común a todos los objetos que puede producir la clase creadora y sus subclases.
- 2.Los **Productos Concretos** son distintas implementaciones de la interfaz de producto.
- 3.La clase **Creadora** declara el método fábrica que devuelve nuevos objetos de producto. Es importante que el tipo de retorno de este método coincida con la interfaz de producto.
- 4.Los **Creadores Concretos** sobrescriben el Factory Method base, de modo que devuelva un tipo diferente de producto.



Ejemplo:

```
public class Principal {  
    public static void main(String[] args){  
        String pl;  
        Conexion co = new Conexion();  
        Scanner escribir = new Scanner(System.in);  
        System.out.println("Ingrese el nombre de un proveedor");  
        pl=escribir.next();  
        Fabrica col = co.getConexion(pl);  
        col.conectar();  
        col.desconectar();  
        pl=escribir.next();  
        Fabrica co2 = co.getConexion(pl);  
        co2.conectar();  
        co2.desconectar();  
        pl=escribir.next();  
        Fabrica co3 = co.getConexion(pl);  
        co3.conectar();  
        co3.desconectar();  
    }  
}
```

MetodoFactory (run) X

```
run:  
Ingrese el nombre de un proveedor  
MYSQL  
Se conecto  
Se desconecto  
ORACLE  
Se conecto  
Se desconecto  
SAD  
no se especifico un proveedor
```


Pros y contras

- Pros
- Evitas un acoplamiento fuerte entre el creador y los productos concretos.
- *Principio de responsabilidad única.* Puedes mover el código de creación de producto a un lugar del programa, haciendo que el código sea más fácil de mantener.
- *Principio de abierto/cerrado.* Puedes incorporar nuevos tipos de productos en el programa sin descomponer el código cliente existente.
- Contras
- Puede ser que el código se complique, ya que debes incorporar una multitud de nuevas subclases para implementar el patrón. La situación ideal sería introducir el patrón en una jerarquía existente de clases creadoras.