Patrones de Diseño: Ejercicios. Patrones de Creación

En una empresa de juguetes se están planteando construir tres clases de juguetes: coches, aviones y naves espaciales con acciones de animación asociadas a su encendido y apagado. Estas acciones serán diferentes para cada juguete. Todos los juguetes tendrán un nombre. ¿Qué patrón aplicarías para facilitar la creación de los distintos tipos de juguetes?

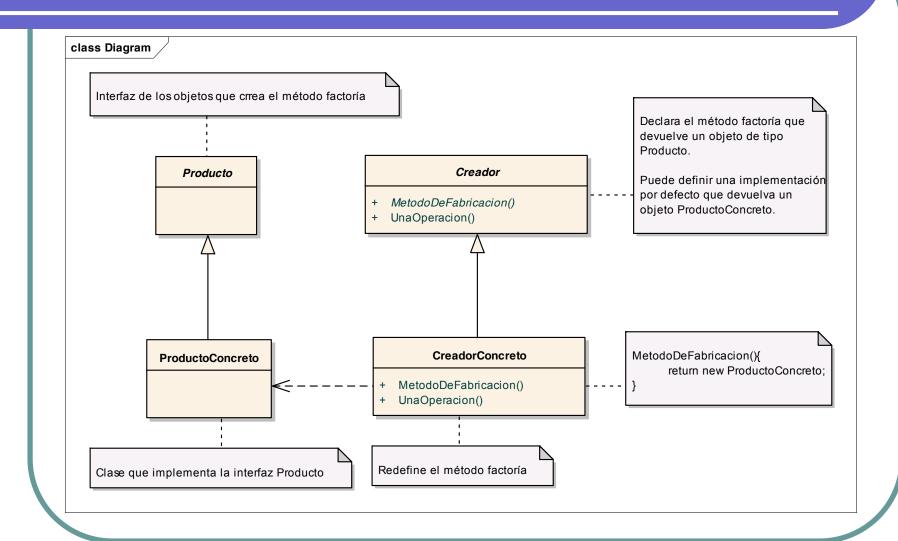




- El patrón que mejor se adapta a este ejercicio es el Factory Method.
 - Define una interfaz para crear un objeto pero deja que sean las subclases quienes decidan que clase instanciar. Permite que una clase delegue en sus subclases la creación del objeto.
 - Nota: También se podría aplicar el patrón Builder.



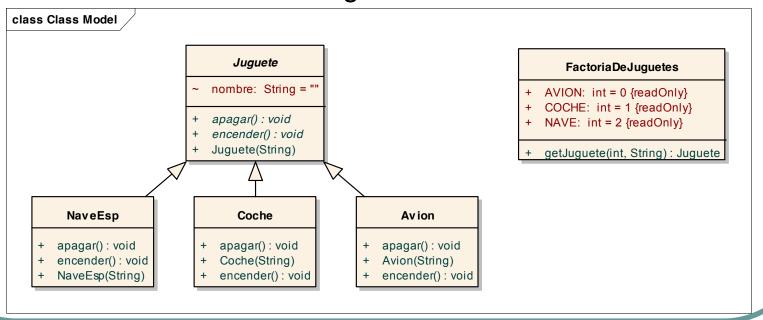








- Identificamos a continuación los elementos del patrón:
 - Producto: Juguete.
 - ProductoConcreto: NaveEsp, Coche, Avion.
 - Creador: FactoriaDeJuguetes.







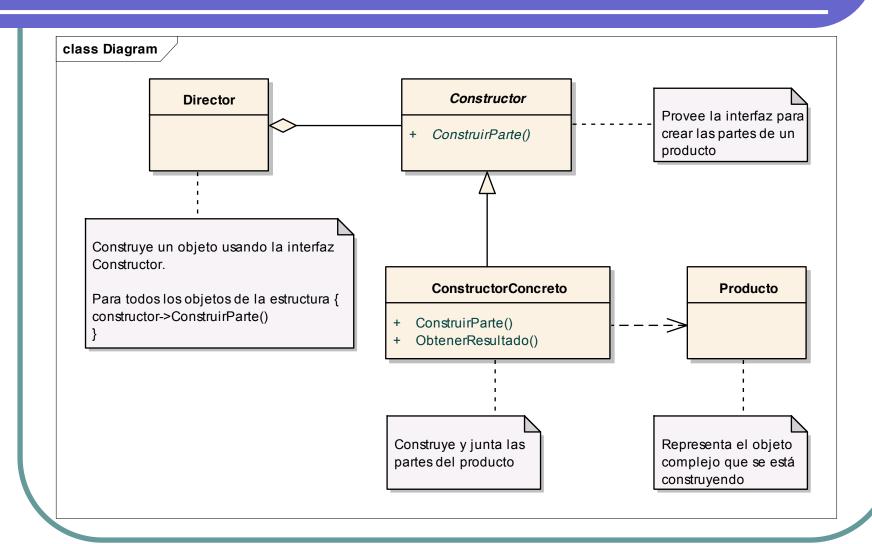
Un estudio de arquitectura pretende realizar una aplicación que le ayude en el diseño de sus inmuebles. En principio el estudio diseña dos tipos de inmuebles: apartamentos y chalets, pero la aplicación debe poder incorporar fácilmente el diseño de otros inmuebles. Cada inmueble se diseña de forma similar, diseñando las habitaciones, el salón, la cocina, los cuartos de baño y la terraza. Todo inmueble tiene una superficie y habitaciones y los cuartos de baño un nombre. ¿Qué patrón aplicarías para facilitar el diseño de los distintos tipos de inmuebles?





- El patrón que mejor se adapta a este ejercicio es el Builder.
 - Separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones.





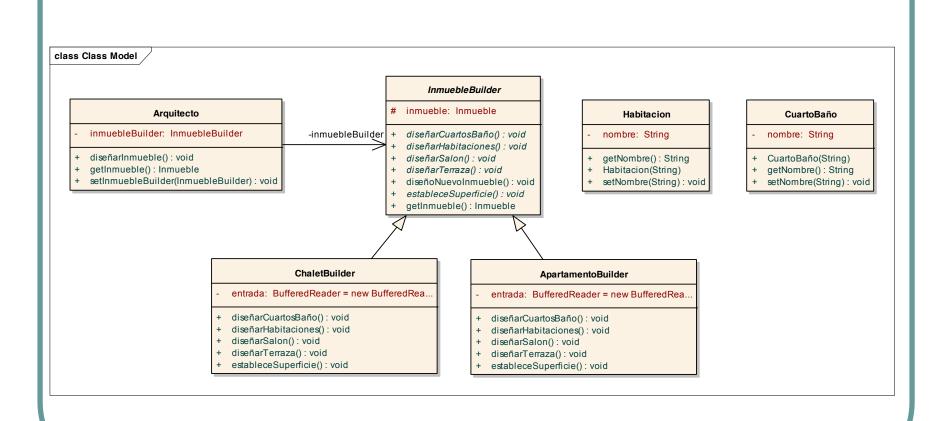




- Identificamos a continuación los elementos del patrón Builder:
 - Constructor: InmuebleBuilder.
 - Constructor concreto: ChaletBuilder, ApartamentoBuilder.
 - Producto: Inmueble.
 - Director: Arquitecto.











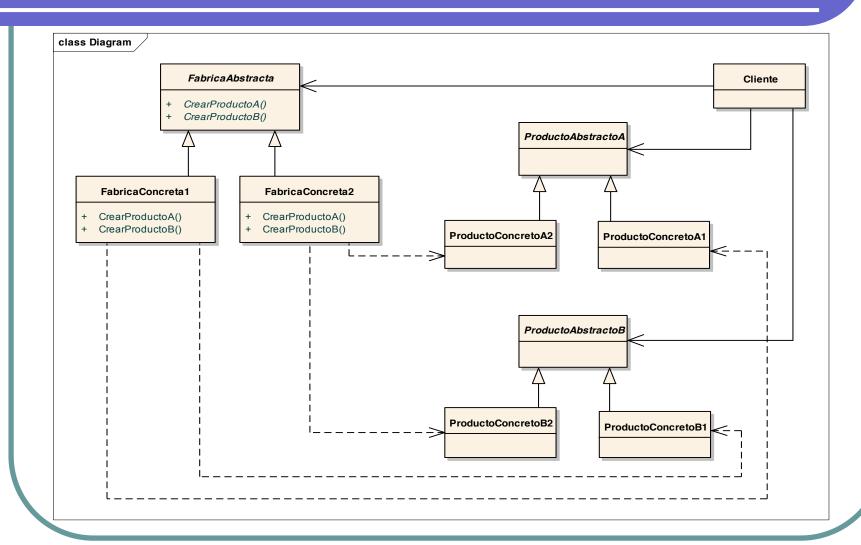
Se desea realizar una aplicación que simule un restaurante donde se sirven distintos tipos de menús. Los menús están compuestos por primer plato, segundo y postre y dentro de cada uno podremos elegir entre distintos tipos de alimentos. ¿Cómo representarías la estructura de clases para poder crear dos tipos de menús y que patrón utilizarías?





- El patrón que mejor se adapta a este ejercicio es el Abstract Factory.
 - Proporciona una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.





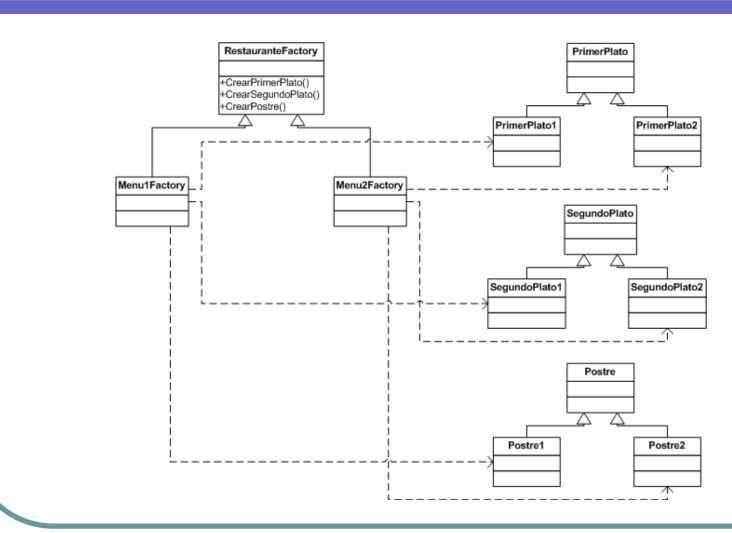




- Identificamos a continuación los elementos del patrón Abstract Factory:
 - FabricaAbstracta: RestauranteFactory.
 - FabricaConcreta: Menu1Factory, Menu2Factory.
 - ProductoAbstracto: PrimerPlato, SegundoPlato, Postre.
 - ProductoConcreto: PrimerPlato1, PrimerPlato2, SegundoPlato1, SegundoPlato2, Postre1, Postre2.









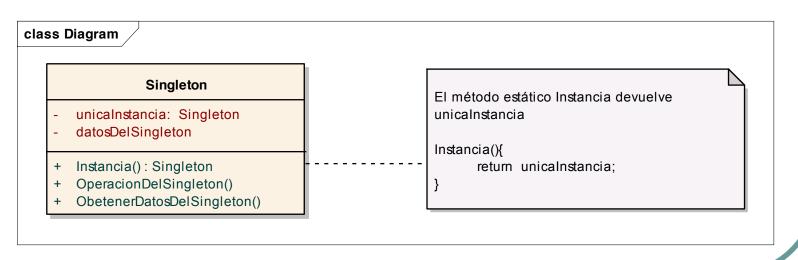


 Tenemos un fichero de configuración cuyos valores pueden ser usados desde cualquier parte de la aplicación. ¿Qué patrón aplicarías para facilitar el acceso a estos valores?





- El patrón que mejor se adapta a este ejercicio es el Singleton.
 - Garantiza que una clase solo tenga una instancia, y proporciona un punto de acceso global a ella. Todos los objetos que utilizan una instancia de esa clase usan la misma instancia.







class Singleton

ConfiguracionEntorno

- instancia: ConfiguracionEntorno
- + getInstancia(): ConfiguracionEntorno
- + getPropiedad(String): String





• Hemos de crear una aplicación para la consulta de datos almacenados en una base de datos de histórico. Sabemos que el volumen de datos que se devuelve en cada consulta y el coste del acceso a base de datos son altos. ¿Qué patrón aplicarías para gestionar los distintos tipos de ordenación de los datos devueltos por la consulta?





- El patrón que mejor se adapta a este ejercicio es Propotype.
 - Se realiza una primera lectura de base de datos. En caso de necesitarse una ordenación diferente a la inicial, se clona el resultado de la primera lectura y se ordena. De esta forma evitamos un segundo acceso a base de datos.



