

Patrón de diseño

Abstract Factory

Alumno: Mondino Facundo

2021

DEFINICIÓN:

Abstract Factory es un patrón de diseño de tipo creacional, ya que permite producir, a través de una interfaz, una familia de objetos que están relacionados o que dependen entre sí, sin tener que definir la clase específica del objeto que se desea crear.

Es por eso que se trabaja con el concepto de abstracción con el objetivo de que el cliente o quien haga uso del patrón pueda interactuar solo con las interfaces abstractas de los objetos creados sin conocer sus clases concretas, las relaciones entre ellos son determinadas por la interfaz abstracta (**fábrica abstracta**).



TERMINOLOGÍA Y CONCEPTOS CLAVES:

- ❖ Cuando se habla de **interfaz** se hace referencia a una clase abstracta que dispone, únicamente, los métodos de creación de objetos abstractos.
- ❖ Cuando se menciona **objeto** se hace referencia a producto.
- ❖ Cuando se habla de una **familia** de objetos se esta representando a un conjunto de productos que disponen de alguna característica en común que hace que se relacionen o bien, que dependan entre sí.
- ❖ La interfaz representa a la **fábrica abstracta**.
- ❖ Se llama **clase concreta** al producto específico o concreto que hereda las características del producto abstracto creado.

USO DEL PATRÓN:

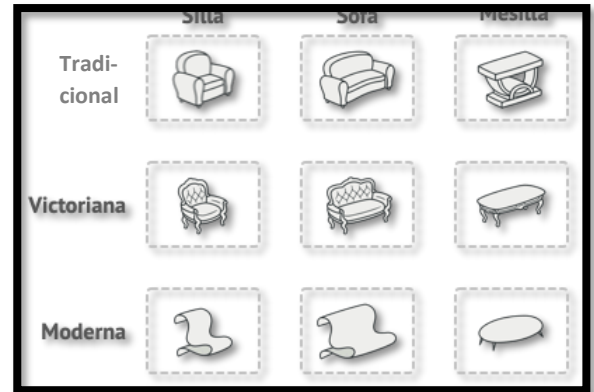
El mismo se utiliza cuando se dispone de una situación problemática que se desarrolla en un escenario donde se requiere **trabajar en conjunto con varios objetos** que están relacionados por medio de alguna característica específica.

Además de brindarle una fábrica abstracta, otorga la posibilidad de pensar a futuro ya que permite agregar nuevas familias de productos a la estructura actual con la que se cuenta.

EJEMPLO CONCEPTUAL:

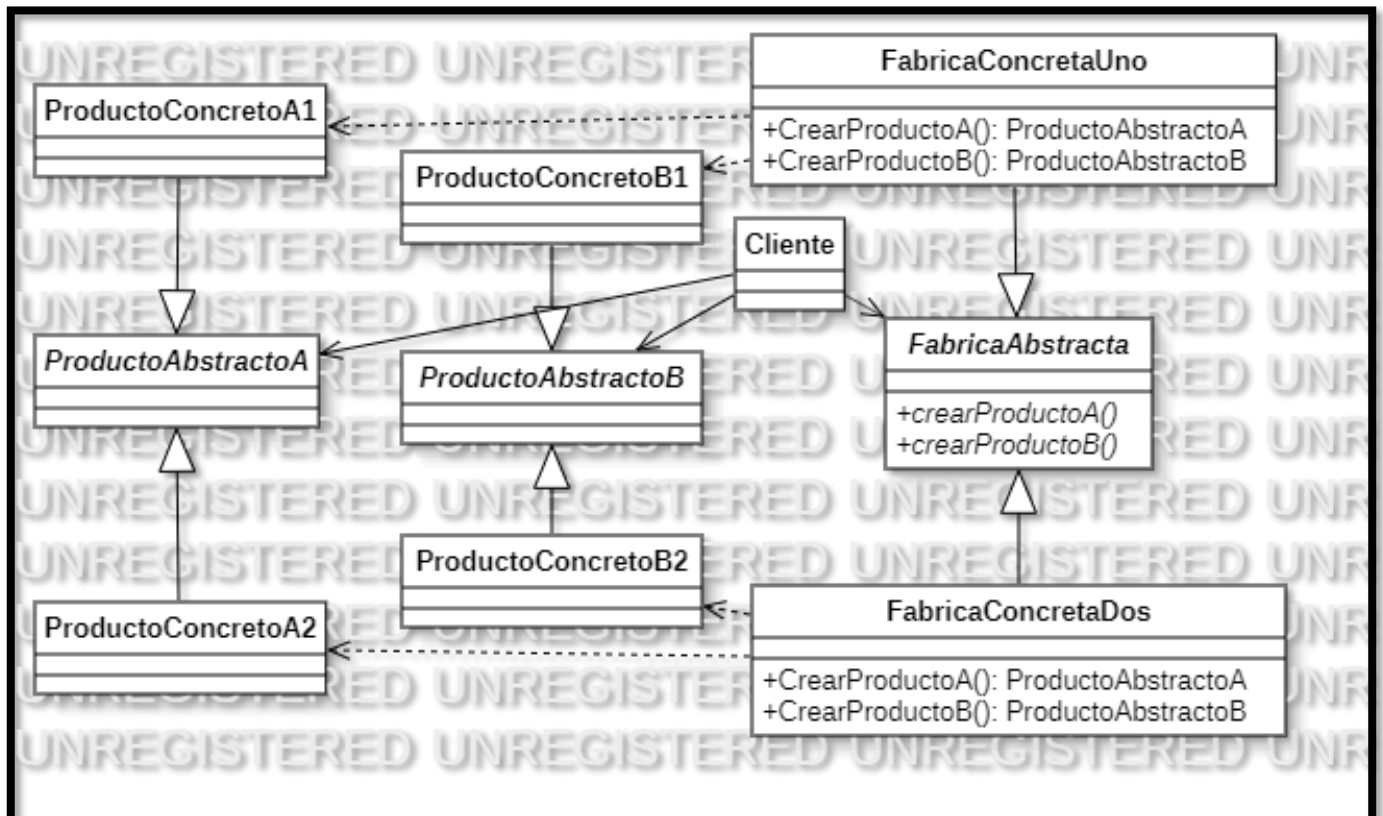
A continuación, se desarrollará un breve ejemplo conceptual para entender los conceptos teóricos mencionados hasta el momento.

Se tiene como escenario del problema, una mueblería. La misma vende muebles a sus clientes, tales como sillas, sillones y mesas ratonas, que vienen en diferentes variantes de **estilo de diseño** como moderno, tradicional y victoriano.



- ❖ Los productos vienen dados por Silla, Sillón y Mesa Ratona.
- ❖ Existen tres familias de productos:
 - Primera familia: Silla Moderna, Sillón Moderno, Mesa Ratona Moderna.
 - Segunda familia: Silla Tradicional, Sillón Tradicional, Mesa Ratona Tradicional.
 - Tercera familia: Silla Victoriana, Sillón Victoriano, Mesa Ratona Victoriana.

ESTRUCTURA:



Se observa que la estructura del patrón está formada por una **fábrica abstracta**, **fabricas concretas**, **productos abstractos y productos concretos**, estos son los elementos (clases) del patrón. A continuación, se comenta en que consiste cada uno de ellos:

- ❖ **FabricaAbstracta** es la interfaz o clase abstracta que dispone, únicamente, los métodos de creación de los productos abstractos, *ProducoAbstractoA* y *productoAbstractoB*. Permite generar o producir los mismos.
- ❖ **ProducoAbstractoA** y **productoAbstractoB** declaran interfaces para las familias de productos distintos pero relacionados.
- ❖ **ProductoConcretoA1**, **ProductoConcretoA2**, **ProductoConcretoB1** y **ProductoConcretoB2** son distintas implementaciones de *ProducoAbstractoA* y *productoAbstractoB* que están agrupados por tipo o variante. Heredan las características de las clases abstractas *ProducoAbstractoA* y *productoAbstractoB*.
- ❖ **FabricaConcretaUno** y **FabricaConcretaDos** generan las familias de productos concretos que se utilizaran en conjunto. Además, son aquellas que implementan (heredan) los métodos de creación definidos en la *FabricaAbstracta*. Cada una trabaja únicamente con un solo tipo o variante de producto, por ende, solo crea dichas variantes de los productos.
- ❖ **Cliente**, es quien representa al usuario o aquel que haga uso del patrón y código, trabaja conociendo únicamente la *FabricaAbstracta*, *ProductoAbstractoA* y *ProductoAbstractoB* y como consecuencia conocerá *ProductoConcretoA1*, *ProductoConcretoA2*, *ProductoConcretoB1* y *ProductoConcretoB2*.

VENTAJAS Y DESVENTAJAS:

Algunas de las **ventajas** más importantes, que surgen con la implementación del patrón son:

- ❖ **Flexibilidad**, ya que se aíslan las clases concretas de los productos, es decir cómo se mencionó anteriormente, el cliente trabaja únicamente con la *FabricaAbstracta*, *ProductoAbstractoA* y *ProductoAbstractoB*.
- ❖ **Escalabilidad**, ya que cuando se requiere agregar una nueva familia de productos a la estructura actual con la que se cuenta, se puede lograr sin tener que realizar cambios complejos en las clases ya creadas. En este caso solo se

deberá agregar una *FabricaConcretaTres*, *ProductoConcretoC1* y *ProductoConcretoC2*.

- ❖ **Veracidad**, ya que los datos que se obtienen de una fábrica concreta son veraces, es decir, le proporcionan al usuario la seguridad de que está trabajando con productos que están relacionados o que son compatibles entre sí.

Como **desventaja** se consideró:

- ❖ Resulta difícil la tarea de **agregar un nuevo producto a la estructura existente**, ya que en este caso se tendrían que modificar varias clases existentes tales como *FabricaAbstracta*, *FabricaConcretaUno*, *FabricaConcretaDos* y además se deberán agregar nuevas clases como *ProductoAbstractoC*, *ProductoConcretoC1* y *ProductoConcretoC2*.

ANÉXO:

Este patrón también recibe el nombre de **fábrica de fábricas**, ya que trabaja con otros patrones de diseño como por ejemplo Factory Method.

Existen varias características que los diferencian, entre ellas la más importante es el **número de objetos** con los que trabajan.

- ❖ Factory Method produce un objeto, es decir tiene **un solo método de creación** con el que trabaja.
- ❖ Abstract Factory produce una familia de objetos, es decir dispone de **dos o más métodos de creación** con los que trabaja.

Para finalizar con la explicación del patrón, se realizó un **ejemplo codificado en c#**, donde se puede ver la manera en la que trabaja el patrón, los elementos del mismo y las relaciones entre ellos, todo desarrollado en un escenario determinado, que en este caso se trata de un **concesionario de autos**.