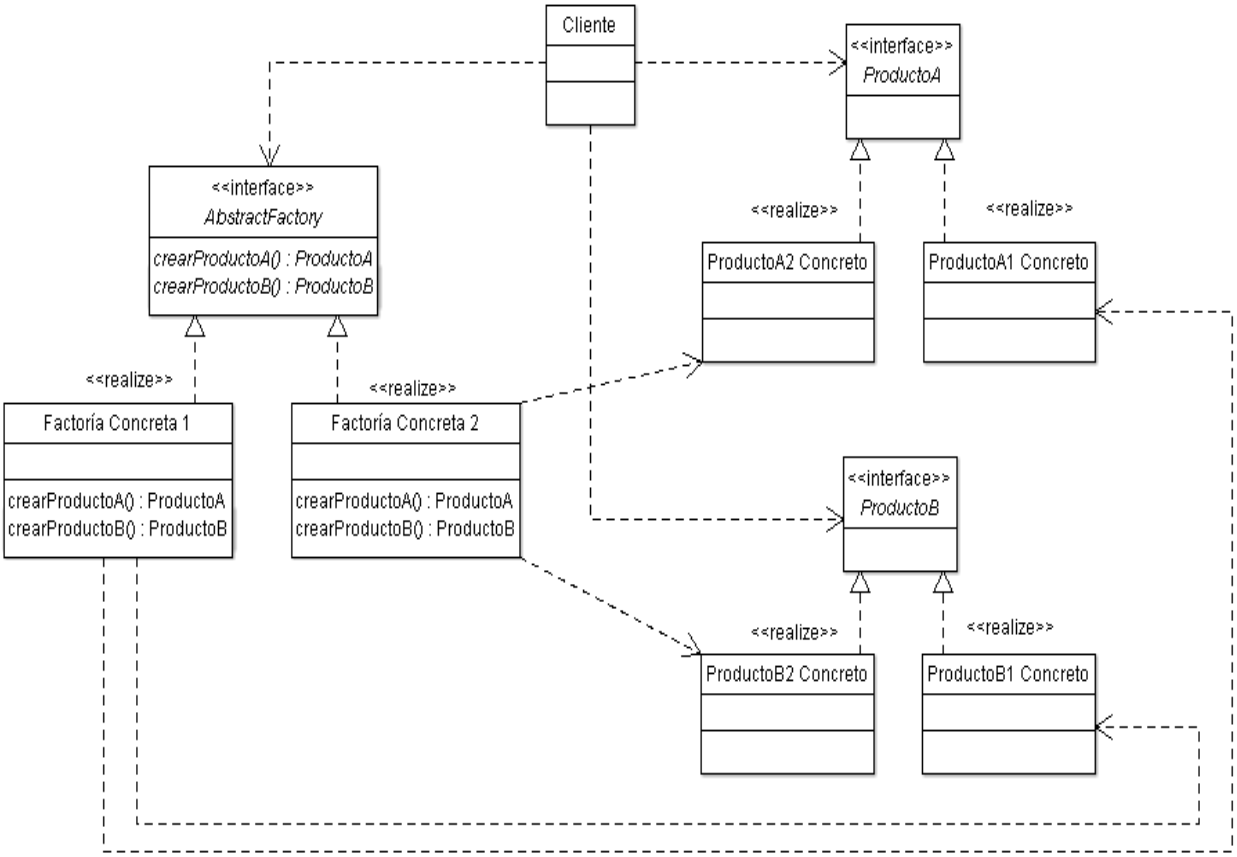
Abstract Factory Method

Contexto y problema

Contexto: Debemos crear diferentes objetos, todos pertenecientes a la misma familia. Por ejemplo: las librerías para crear interfaces gráficas suelen utilizar este patrón y cada familia sería un sistema operativo distinto. Así pues, el usuario declara un Botón, pero de forma más interna lo que está creando es un BotónWindows o un BotónLinux, por ejemplo.
El problema que intenta solucionar este patrón es el de crear diferentes familias de objetos.



La estructura típica del patrón Abstract Factory es la siguiente:
Cliente: La clase que llamará a la factoría adecuada ya que necesita crear uno de los objetos que provee la factoría, es decir, Cliente lo que quiere es obtener una instancia de alguno de los productos (ProductoA, ProductoB).

AbstractFactory: Es la definición de la interfaces de las factorías. Debe de proveer un método para la obtención de cada objeto que pueda crear. ("crearProductoA()" y "crearProductoB()")
Factorías Concretas: Estas son las diferentes familias de productos. Provee de la instancia concreta de la que se encarga de crear. De esta forma podemos tener una factoría que cree los elementos gráficos para Windows y otra que los cree para Linux, pudiendo poner fácilmente (creando una nueva) otra que los cree para MacOS, por ejemplo.
Producto abstracto: Definición de las interfaces para la familia de productos genéricos. En el diagrama son "ProductoA" y "ProductoB". En un ejemplo de interfaces gráficas podrían ser todos los elementos: Botón, Ventana, Cuadro de Texto, Combo... El cliente trabajará directamente sobre esta interfaz, que será implementada por los diferentes productos concretos.
Producto concreto: Implementación de los diferentes productos. Podría ser por ejemplo "BotónWindows" y "BotónLinux". Como ambos implementan "Botón" el cliente no sabrá si está en Windows o Linux, puesto que trabajará directamente sobre la superclase o interfaz.
Supongamos que disponemos de una cadena de pizzerías. Para crear pizzas disponemos de un método abstracto en la clase Pizzería que será implementada por cada subclase de Pizzería.
abstract Pizza crearPizza()
Concretamente se creará una clase PizzeríaZona por cada zona, por ejemplo la Pizzería de New York sería PizzeriaNewYork y la de California PizzeríaCalifornia que implementarán el método con los ingredientes de sus zonas.
Las pizzas son diferentes según las zonas. No es igual la pizza de New York que la pizza de California. Igualmente, aunque usarán los mismos ingredientes (tomate, mozzarella...) no los obtendrán del mismo lugar, cada zona los comprará donde lo tenga más cerca. Así pues podemos crear un método creador de Pizza que sea Pizza(FactoriaIngredientes fi);
Como vemos utilizamos la factoría abstracta (no las concretas de cada zona, como podría ser IngredientesNewYork o IngredientesCalifornia). Pizza podrá obtener los ingredientes de la factoría independientemente de donde sea. Sería fácil crear nuevas factorías y añadirlas al sistema para crear pizzas con estos nuevos ingredientes. Efectivamente, en este ejemplo cliente es Pizza y es independiente de la Factoría usada.
El creador de la Pizza será el encargado de instanciar la factoría concreta, así pues los encargados de instanciar las factorías concretas serán las pizzerías locales. En PizzeríaNewYork podemos tener el método crearPizza() que realice el siguiente trabajo:

```
Pizza crearPizza() {
   FactoríaIngredientes fi = new IngredientesNewYork();
   Pizza pizza = new Pizza(fi); // Uso de la factoría
   pizza.cortar();
   pizza.empaquetar();
   return pizza;
}
```

Como conclusión podemos observar que gracias a la factoría de ingredientes crear una nueva zona, por ejemplo una pizzería en Barcelona, no nos implicaría estar modificando el código existente, solo deberemos extenderlo (uno de los pilares de la Ingeniería del software) ya crearíamos la subclase de Pizzería: PizzeríaBarcelona que al instanciar la factoría solo debería escoger la factoría de Barcelona. Obviamente se debería crear la factoría de Barcelona que se encargaría de crear los productos obtenidos de Barcelona. Así que en ningún momento modificamos las pizzerías existentes, la superclase pizzería o las otras factorías o productos, solo creamos nuevas clases.