Esteban Alvarado – 202125000

**Taller 5**

1. **Información general del proyecto: para qué sirve, cuál es la estructura general del diseño, qué grandes retos de diseño enfrenta (i.e. ¿qué es lo difícil?). Deben incluir la URL para consultar el proyecto:**

**URL:** [**https://github.com/AchillesKastanas/Computer-Store-DP.git**](https://github.com/AchillesKastanas/Computer-Store-DP.git)

* El proyecto "Computer-Store-DP" simula una tienda de computadoras que utiliza el patron “Factory” para crear y gestionar diferentes tipos de computadoras y clientes. El proyecto crea computadores y se los asigna a los clientes de manera aleatoria, con distintas especificaciones y tipos de computadores.
* Estructura del Diseño: El proyecto utiliza principalmente el patrón Factory para la creación de computadoras. Este patrón se implementa a través de una interfaz IComputerFactory y clases concretas como DesktopPCFactory y LaptopFactory.
* Retos de Diseño: El principal desafío es la fácil adición de nuevos tipos de computadoras sin alterar significativamente el código existente.

1. **Información y estructura del fragmento del proyecto donde aparece el patrón. No se limite únicamente a los elementos que hacen parte del patrón: para que tenga sentido su uso, probablemente va a tener que incluir elementos cercanos que sirvan para contextualizarlo.**

* **Patrón Factory:** Se implementa mediante las siguientes clases e interfaces:
* **IComputerFactory:** Una interfaz que define el método createComputer.
* **DesktopPCFactory y LaptopFactory:** Clases concretas que implementan IComputerFactory, proporcionando la lógica para crear diferentes tipos de computadoras.
* **ComputerFactory:** Una clase que utiliza las fábricas concretas para crear objetos de computadora.

Este patrón se utiliza para crear diferentes tipos de computadoras, permitiendo que la tienda maneje fácilmente la variedad de productos sin depender de clases concretas.

1. **Información general sobre el patrón: qué patrón es y para qué se usa usualmente:**

El método “Fabric” es un patrón de diseño creacional que resuelve el problema de crear objetos de producto sin especificar sus clases concretas. El Método de Fábrica define un método, que debería utilizarse para crear objetos en lugar de usar una llamada directa al constructor

1. **Información del patrón aplicado al proyecto: explicar cómo se está utilizando el patrón dentro del proyecto:**

El patrón Factory se utiliza para crear instancias de DesktopPC y Laptop. Esto se hace a través de ComputerFactory, que encarga la creación del objeto a las fábricas concretas (DesktopPCFactory y LaptopFactory), permitiendo así que la tienda maneje diferentes tipos de computadoras de manera flexible.

Diagrama

Descripción generada automáticamente

Esto hace que al programa solo le importe que el objeto se cree, pero no como.

1. **¿Por qué tiene sentido haber utilizado el patrón en ese punto del proyecto? ¿Qué ventajas tiene?**

Permite mantener la creación de todos los objetos en un solo lugar y evitar la dispersión de la palabra clave 'new' por toda la base de código. Permite escribir código con bajo acoplamiento y con un código fácil de entender.

1. **¿Qué desventajas tiene haber utilizado el patrón en ese punto del proyecto?**

Una desventaja es que se vuelve más difícil de entender el código para los desarrolladores, se vuelve más complicado el código y introduce varias clases e interfaces.

1. **¿De qué otras formas se le ocurre que se podrían haber solucionado, en este caso particular, los problemas que resuelve el patrón?**

En lugar de usar fábricas concretas, se podrían haber utilizado métodos para crear diferentes tipos de computadoras, así no se debería haber utilizado el método “Factory”, y así es posible resolver el mismo problema, con una mirada distinta.