README

Juan Aurelio Juarez Ubaldo - 421095568
Sergio Mejia Caballero - 3151149

1 Teoria

1. Menciona los principios de diseño esenciales de los patrones Factory, Abstract Factory y Builder. Menciona una desventaja de cada patron.

a) Factory. Este patron nos ayuda a crear objetos de una superclase, mientras que las subclases pueden alterar el tipo de objetos que se crearan. Tambien podemos agregar o remover nuevos tipos sin modificar el codigo existente. Una desventaja del patron es que debemos introducir varias interaces y clases nuevas al proyecto.

b) Abstract Factory. Este patron nos ayuda a crear familias de objetos relacionadas sin tener que especificar sus clases concretas. Una desventaja es que debemos agregar varias interfaces nuevas, la interfaz abstractFactory y una interfaz por cada "familia" de objetos que tengamos, por lo que el codigo puede complicarse.

c) Builder. Este patron nos permite crear objetos complejos paso por paso, podemos crear distintas representaciones del objeto usando el mismo codigo constructor. Una desventaja, es que si solo tendremos uno o dos constructores diferentes, la complejidad añadida por el patron puede ser excesiva, por lo que es mejor si sabemos que tendremos varias opciones de construcción. Observamos que tienen desventajas similares, ya que en todas debemos agregar gran cantidad de interfaces y subclases al codigo, sin embargo, esta complejidad añadida nos ayuda a tener un codigo mas limpio y extensible.

2 README

Para esta practica, simulamos la creacion de carros en los que el usuario puede elegir entre distintos tipos de componentes para sus carros, Decidimos usar Abstract Factory ya que estamos eligendo componentes de una "familia" de objetos (Llantas, Motor, etc) y creemos que el patron que nos iba a permitir tener mas flexibilidad en cuanto a la creacion de estos componentes fue abstract

factory. Ademas, creamos una clase llamada ComponenteCarro para evitar escribir en cada componente sus atributos de ataque, defensa, velocidad y costo. En nuestra simulacion, el usuario puede elegir entre un carro ya predeterminado o crear un carro personalizado donde cada que el usuario pide un componente, se crea el componente concreto, y si ya eligio todos los componentes, se crea el objeto de tipo Auto y se realiza el cobro, si el dinero del cliente no es suficiente, entonces el cobro no se realiza.