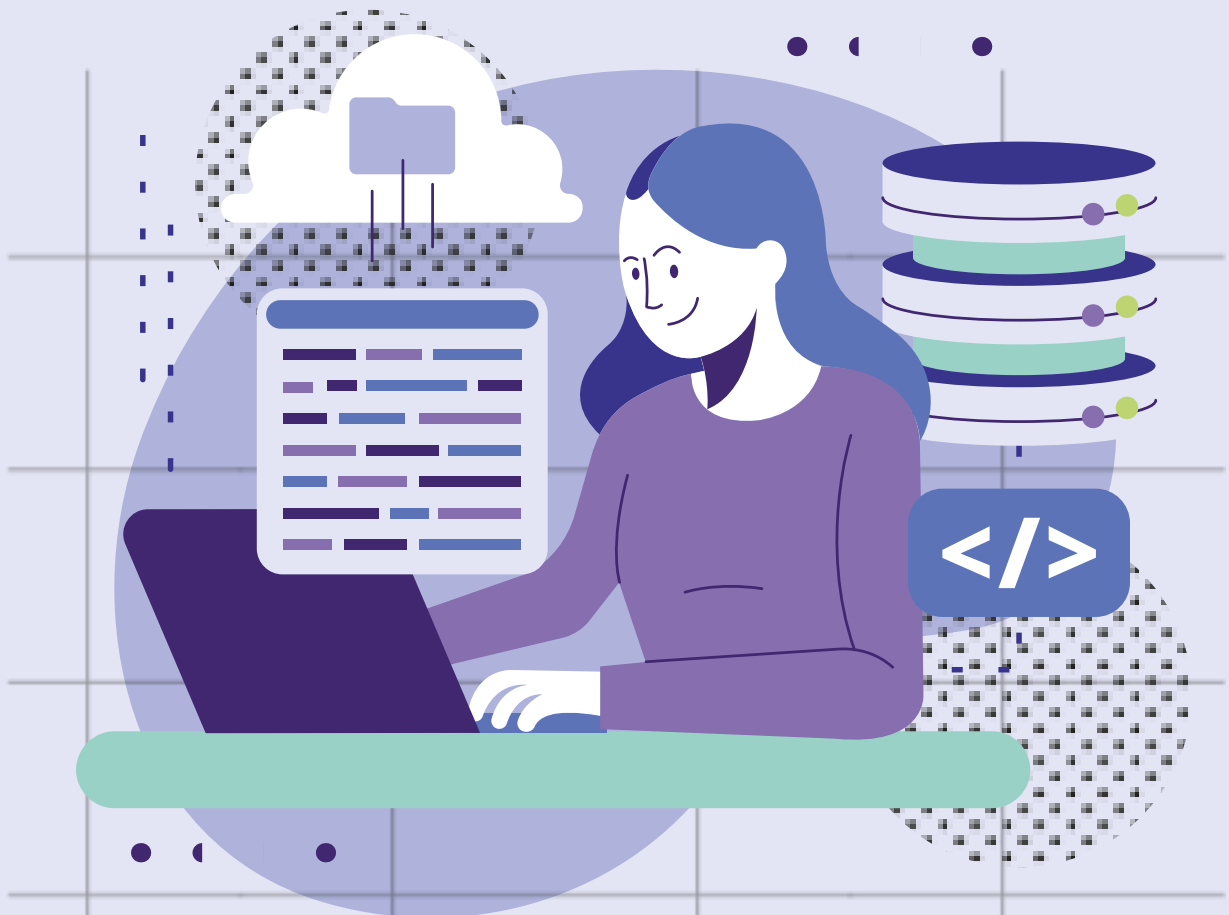


Patrones de Diseño



Patrón de Diseño Factory

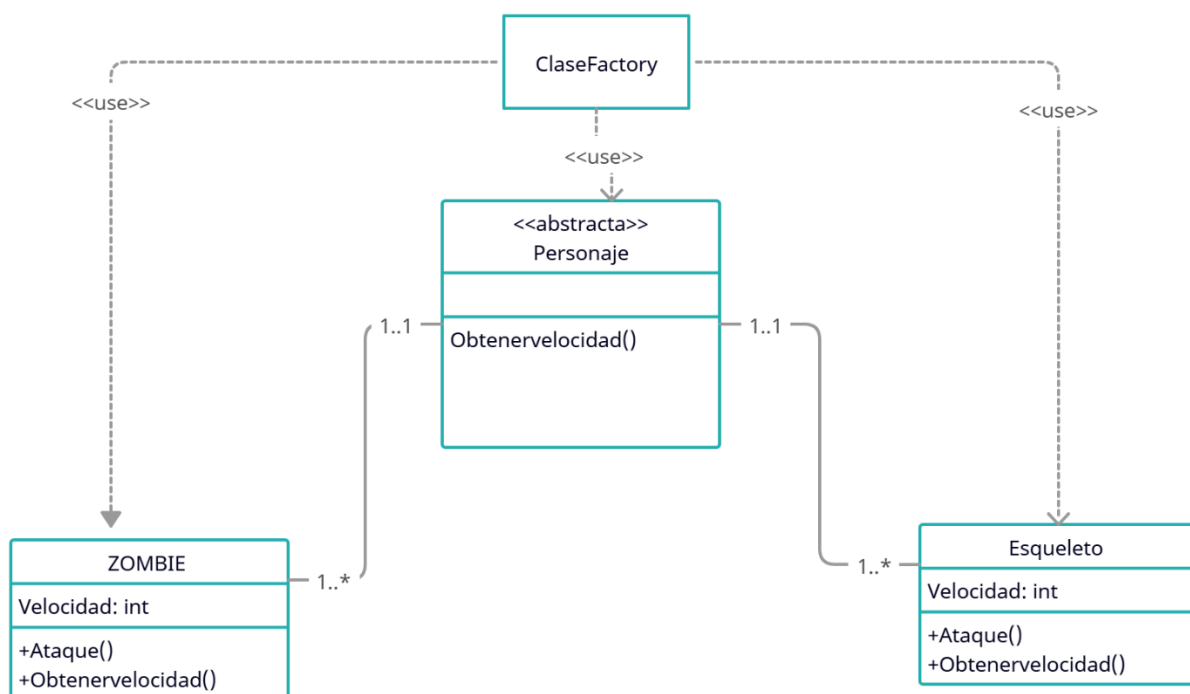
Es un patrón de diseño creacional que proporciona una interfaz para crear objetos en una superclase, mientras permite a las subclases alterar el tipo de objetos que se crearán.

Ejercicio 1:

Crear un programa que contenga dos personajes: "Esqueleto" y "Zombi". Cada personaje tendrá una lógica diferente en sus ataques y velocidad. La creación de los personajes dependerá del nivel del juego:

- En el nivel fácil se creará un personaje "Esqueleto".
- En el nivel difícil se creará un personaje "Zombi".

Debes aplicar el patrón de diseño Factory para la creación de los personajes.

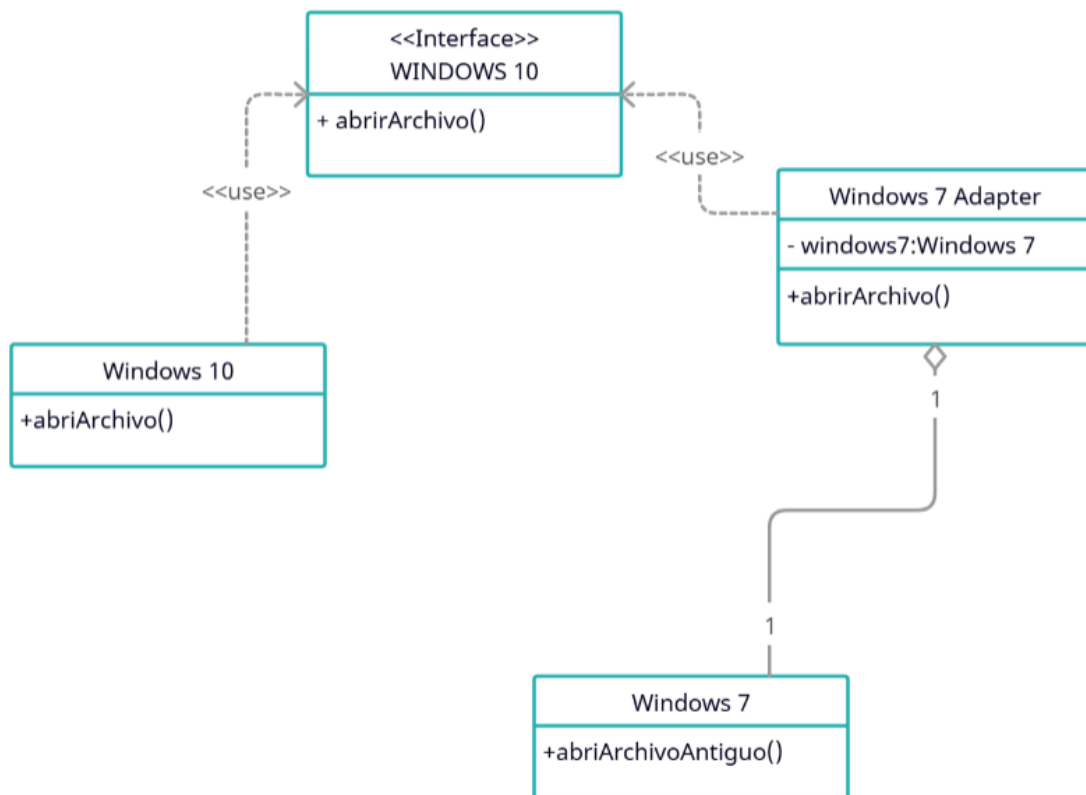


Patrón de Diseño Adapter

es un patrón de diseño estructural que permite la colaboración entre objetos con interfaces incompatibles.

Ejercicio 2:

Estamos trabajando con distintas versiones de sistemas operativos Windows 7 y Windows 10. Al compartir archivos como Word, Excel, Power Point, surgen problemas al abrirlos en Windows 10 debido a la falta de compatibilidad con la versión Windows 7. Debes crear un programa donde Windows 10 pueda aceptar estos archivos independientemente de que sean de versiones anteriores.



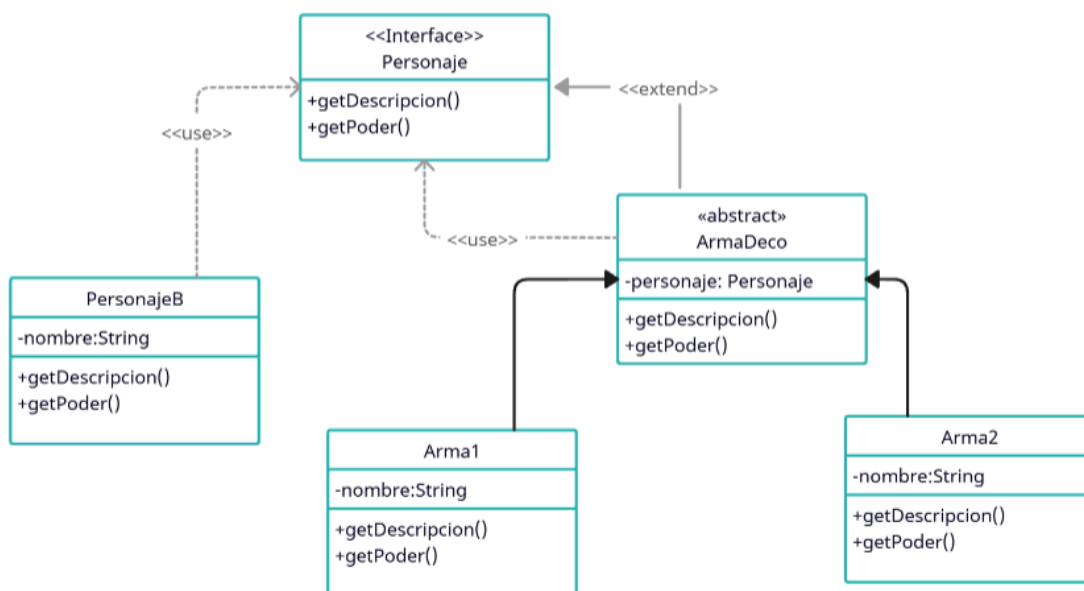
Patrón de Diseño Decorator

Es un patrón de diseño estructural que te permite añadir funcionalidades a objetos colocando estos objetos dentro de objetos encapsuladores especiales que contienen estas funcionalidades.

Ejercicio 3:

Crear un programa donde sea posible añadir diferentes armas a ciertos personajes de videojuegos. Debes utilizar al menos dos personajes para este ejercicio

Patron de diseño DECORATOR



Patrón de Diseño Strategy

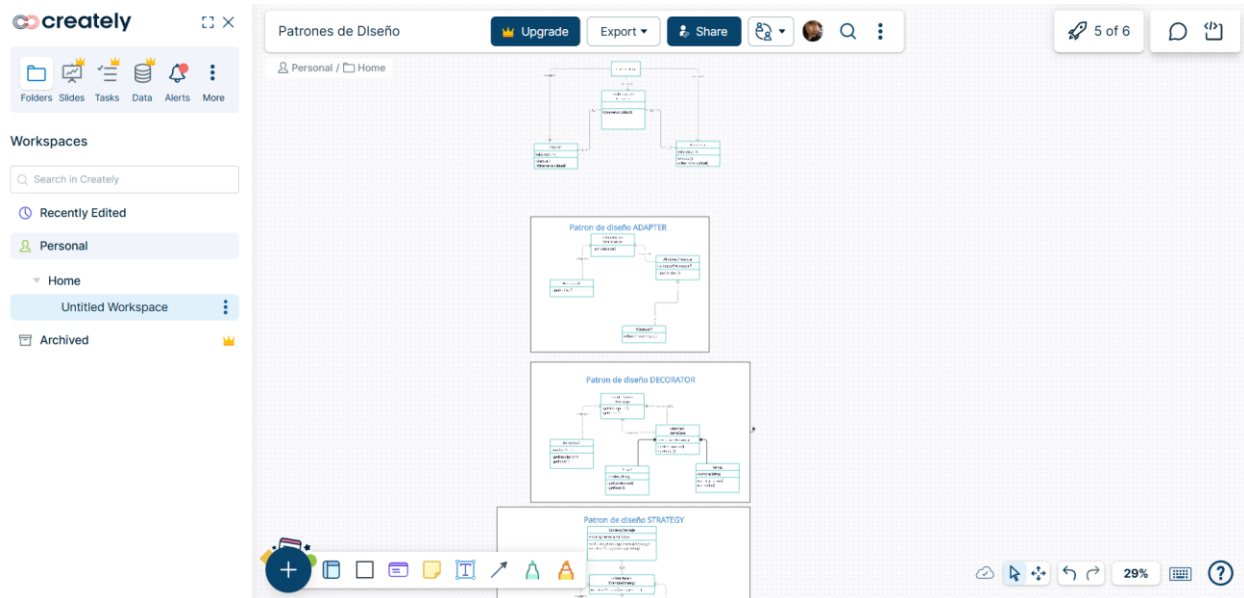
Es un patrón de diseño de comportamiento que te permite definir una familia de algoritmos, colocar cada uno de ellos en una clase separada y hacer sus objetos intercambiables.

Ejercicio 4:

Tenemos un sistema donde mostramos mensajes en distintos tipos de salida, como por consola, formato JSON y archivo TXT. Debes crear un programa donde se muestren todos estos tipos de salidas.



Programa donde se realizaron los diagramas



Referencias

[Factory Method](#)

[ABSTRACT FACTORY | PATRONES de DISEÑO](#)

[Adapter](#)

[Cómo implementar el Patrón ADAPTER?](#)

[Strategy](#)