Sprint 1

Jessica Tatiana Gaitán Yusti

2 patrones de diseño explicados con 2 ejemplos de la vida real y con código fuente

1. Patrón de diseño Decorator

Genera capas a una información base para facilitar diferentes formas de acceso sin necesidad de modificar la base ni sobrecargar el programa.

Primer ejemplo: una notificación que se puede enviar a texto, fb o whatsapp que con decorator se puede usar en las 3 o en 2 según elija el usuario.

Ejemplo: una encuesta, que arroja los datos más relevantes recogidos según que datos se desea cotejar, así se ingresa en la base los datos de la encuesta, y en los wrappers o decoradores se generan los accesos a los datos mas relevantes y se pueden cotejar según la necesidad del investigador.

Código fuente:

```
interface encuesta is
    method ingresar(data)
    method leer():data

class datosencuesta implements encuesta is
    constructor datosencuesta(filename) { ... }

    method ingresar(data) is

    method leer():data is

class decoradorencuesta implements datosencuesta is
    protected field wrappee: encuesta

    constructor decoradorencuesta(source: encuesta) is
        wrappee = source

        method ingresar(data) is
        wrappee.ingresar(data)


    method leer():data is
        return wrappee.leer()
```

```
.
class EncryptionDecorator extends decoradorencuesta is
    method ingresar(data) is


    method leer():data is


class CompressionDecorator extends decoradorencuesta is
    method leer(data) is


    method leer():data is


class analisis is
    method analisisdatos() is
        source = new encuesta("somefile.dat")
        source.ingresar(datos personales)


        source = new CompressionDecorator(source)
        source.ingresar(datos personales)


        source = new EncryptionDecorator(source)



class datospersonales is
    field source: encuesta

    constructor datospersonales(source: encuesta) { ... }

    method load() is
        return source.leer()

    method save() is
        source.ingresar(datospersonales)
```

2. Patrón de diseño Facade

Se crea una fachada para el acceso que esconde los diferentes procesos que se llevan a cabo y que no es necesario que sean accesibles o visibles durante el proceso.

Ejemplo 1: un convertidor de videos, que usa diferentes tipos de procesos para convertir en diferentes programas, solo muestra el video y la opción convertir, mientras las otras clases que se ejecutan decodifican y convierten el audio y la imagen.

Ejemplo 2: comprar en un supermercado , solo se muestra el producto disponible y se agrega al carrito de compras, mientras las clases se encargan del tipo de producto, numero de productos, envío, empaque, pagos, inventario.

Código fuente

```
class supermercado is
    method comprar(producto, agregaralcarrito):File is
        file = new producto(filename)
        inventario = new producto.inventario(file)
        if (inventario == producto)
            producto = disponible()
        else
            inventario = numero product()
        numero_inventario = numeroproducto(producto, agregaralcarrito)
        carrito = numeroproducto()
        inventario1 = (numeroinventario - numeroproducto()
        return inventario1(result)

class comprar is
    method main() is
        comprar = numeroproducto()
        inventario2 = inventario1
        pagar()
```