

# **PATRONES SOFTWARE**

## **LABORATORIO - P7**

### **PATRONES ESTRUCTURALES**

#### **Práctica 7**

- **Ejercicio 1:**

Se desea crear una aplicación para el manejo de TPV (Terminal de Punto de Venta) en diferentes supermercados. Se quiere diseñar la aplicación de tal forma que contemple la aplicación de descuentos al total de una venta. Habrá diferentes políticas de descuentos como:

- "los miércoles un 5% al total de una venta"
- "un 10% en cada compra superior a 150€"
- "descuentos para determinados tipos de clientes", etc.

La aplicación debe permitir la acumulación de distintos descuentos para una misma venta. ¿Cómo podrías diseñar la aplicación utilizando patrones de diseño de forma que la incorporación de nuevos descuentos fuera lo más sencilla posible? ¿Cómo lo implementarías?

- **Ejercicio 2:**

Se desea realizar una aplicación de consulta de datos de los empleados de una empresa. De cada empleado se almacenan: (1) identificador, (2) contraseña, (3) nombre, (4) departamento, y (5) sueldo. Un empleado puede consultar tanto sus propios datos como los de sus compañeros. Como se puede suponer, algunos de estos datos son confidenciales y solo los puede consultar el propio empleado.

Por lo tanto, si un empleado consulta sus datos, se le mostrarán: (1) identificador, (2) nombre, (3) departamento, y (4) sueldo. Si un empleado consulta los datos de otro empleado, se le mostrarán: (1) identificador, (2) nombre, y (3) departamento. ¿Qué patrón(es) de diseño podríamos emplear para resolver este problema? ¿Cómo lo implementarías?

- **Ejercicio 3:**


Se desea realizar una aplicación para consultar los datos de la clientela de una tienda. La tienda emplea una aplicación de facturación donde cada factura también almacena los datos de cada cliente. La nueva aplicación que desarrollemos deberá utilizar los datos de la aplicación de facturación para obtener los datos de los clientes. Cada factura contiene los siguientes datos:

- Factura: (1) Número, (2) Concepto, (3) Importe, y (4) Fecha.
- Cliente: (1) CIF, (2) Nombre, (3) Dirección.

¿Cómo podrías reutilizar la aplicación de facturación para proporcionar los datos de los clientes? ¿Qué patrón(es) utilizarías? ¿Cómo lo implementarías?

- **Ejercicio 4:**

Una clase *TextView* representa un componente gráfico de tipo *VentanaDeTexto*, que es una subclase de *Component*. Esta clase *Component* es la raíz de la jerarquía de clases que representa todos los componentes gráficos. Se desea definir ventanas de texto con diferentes tipos de bordes (3D, plain, padded, etc.) y barras de desplazamiento (horizontal y vertical). La clase *TextView* tiene un método *dibujar*.



```
class TextView {  
    public void dibujar() {  
        // Dibujar TextView Object  
    }  
}
```

Realiza el diagrama de clases resultado de aplicar el patrón Decorator a la jerarquía de clases del problema.