

## **DISEÑO DE JERARQUÍA DE CLASES**

He creado cinco clases relativas al empleado y sus subclases, una clase UT6Problema1 que contiene las clases que interactúan con las anteriores y dos clases, una que contiene la interfaz Gestionable y la otra que contiene la interfaz Evaluable. Creé la clase Main, que contiene las pruebas del sistema y un menú de usuario.

De esta manera separo las clases principalmente metódicas de las que tienen que ver más con la construcción de empleados.

(La realidad es que hice todas las clases en la clase UT6Problema1, pero tuve problemas de jerarquía cuando empecé a hacer el Main, entonces separé las clases Empleado y derivadas para que el programa detectase las clases de manera correcta)

## **HERENCIA, POLIFORMISMO Y INTERFACES**

Utilizo la herencia, de forma que las clases derivadas de Empleado obtienen sus dos métodos abstractos, los cuales instancian cada uno individualmente. También obtienen unos atributos principales y después en algunas subclases instancio atributos individuales y separadas.

Empleo las interfaces Gestionable y Evaluable para instanciarlas individualmente en las clases de empleados correspondientes.

## **HERENCIA VS COMPOSICIÓN**

La herencia hace que, en este sistema, los distintos atributos que habría que repetir en cada "subclase" en la composición, no sean repetidos. De esta manera tenemos un código eficiente y fácil de visualizar.

Por otro lado, personalmente me gusta la simpleza de la composición, siento que el poliformismo y las interfaces añaden muchas capas de complejidad y subjetividad, y lo único bueno que puedo puntualizar, es que cuando terminas un proyecto y hay que mantenerlo, es mejor a la hora de ver y entender los métodos abstractos.

## **OTROS**

El problema más grande que he tenido es el de jerarquizar los archivos incorrectamente, como he escrito antes, y el segundo y único problema a parte de este ha sido instanciar un array con valor -1 en un bucle for, lo arreglé poniendo un if si i era 0, que no hiciese esa instancia, y otro if si i era mayor a 0 con esa instancia igual.