

PROGRAMACIÓN III

DIAGNÓSTICO: Maco Wins

Alumna: Martina Berraz

Profesores:

* Franco Bulgarelli
* Luis Tomas Cannavo

Leyendo el problema planteado lo primero que identifiqué fue la necesidad de una clase *Prenda*. De ella podrían heredar los tipos mencionados: sacos, pantalones y camisas. Como estos son los únicos que se deberían poder vender, dado que no tendría que existir una venta de una prenda que no fuese uno de esos tres tipos, la clase *Prenda* podría definirse como abstracta y contener las operaciones comunes a las clases que de esta heredan.

Dado que cada prenda debía tener su precio, pensé en añadirle una propiedad y que esta sea asignada un valor por defecto en cada una de las subclases (*Camisa, Pantalón, Saco*). Sin embargo, el problema requiere que este se modifique dependiendo del estado especificado de la prenda: nueva, promoción y liquidación. Por lo tanto, pensé en hacer una clase abstracta, llamada *Estado*, de la que podrían heredar los tres mencionados anteriormente. Cada estado tendría su propia implementación de una operación de la clase base que calcule el precio. Así los mismos objetos sabrían calcular su precio una vez identificado su estado. Una alternativa que consideré fue hacer que la clase *Prenda* tenga una propiedad que represente el estado y que simplemente en uno de los métodos de la clase se parta de este para que haciendo uso de condicionales se le asigne un valor al precio. Decidí ir por la primera dado que me resultaba más simple si le delegaba a cada estado la tarea de calcular el precio.

En cuanto a las ventas, dado que sería necesario guardarlas para calcular las ganancias de un determinado día, pensé en crear una clase llamada *Local* que tenga una lista donde guardarlas. En línea con los requerimientos especificados, la clase *Venta* debería contener una lista de prendas y propiedades con la cantidad de prendas vendidas y la fecha. Puesto que el precio de cada venta dependería de la forma de pago que únicamente podía tomar el valor de efectivo o tarjeta pensé en crear una clase abstracta *Tipo* de la que estos dos podrían heredar. Así cada venta le delegaría el cálculo del precio a esta clase y sería más simple implementar el pago con tarjeta. La clase *Efectivo* podría haberse omitido si la clase Tipo no se definiera como abstracta pero me pareció necesario diferenciarlos por si en algún futuro se quisiera realizar algún tipo de acción al identificarse que el cliente utilizó este método de pago.