

Maco Wins

Se requiere:

- Identificar los requerimientos
- Presentar una solución usando el paradigma de objetos (pseudocódigo, diagrama de clases).
- Explicar todo lo que considere necesario en prosa.
- Si se descarta alguna alternativa durante el desarrollo de la solución, o si se tiene otra solución, explicarla brevemente.



La conocida empresa de ropa formal para caballeros, Macowins, es capaz de darle soporte a la venta de prendas. Un fragmento de la grabación del analista con el cliente:

“Queremos saber el **precio de venta de una prenda y sus tipos**, los tipos de prenda son: sacos, pantalones, camisas.”

El **cálculo del precio de una prenda** es, el precio propio de la prenda **modificado según el estado de la prenda**, que pueden ser:

- Nueva: en este caso no modifican el precio base.
- Promoción: Le resta un valor fijo decidido por el usuario.
- Liquidación: Es un 50% del valor del producto.

Ah, un requerimiento más: Macowins **registra las ventas de estas prendas** y **necesita saber las ganancias de un determinado día**.

“Cada venta tiene asociada las prendas que se vendieron, su cantidad y la fecha de venta. **Las ventas pueden ser en efectivo o con tarjeta**. **En el caso que sea con tarjeta**, tienen el mismo comportamiento que en efectivo (el cual no modifica el precio), sólo que **se le aplica un recargo según la cantidad de cuotas seleccionadas** (cantidad de cuotas * un coeficiente fijo + 0.01 del valor de cada prenda).”

Solución:

Requerimientos funcionales:

- Saber el tipo de una prenda.
- Calcular el precio de una prenda en base a su estado.
- Registrar la venta de las prendas.
- Saber las ganancias de un determinado día.
- Poder pagar en efectivo o con tarjeta (en cuotas)
- Calcular y aplicar un recargo al pago en tarjeta en base a la cantidad de cuotas y el valor de las prendas.

Requerimientos no funcionales:

Para implementar un modelo inicial de solución a este problema utilicé Wollok (puede encontrarse el código en el repositorio). Cabe aclarar que la alternativa de solución que diseñé no es la única, sobre lo cual comentaré en el último párrafo.

Existen algunas incertezas (producto de comentarios vagos/ambiguos en el análisis) sobre las cuales tomé ciertas libertades a la hora del diseño. Por ejemplo, cuando una prenda está en promoción resta un valor fijo al precio decidido por el usuario. ¿Ese valor fijo es el mismo para todas las prendas? ¿Para todos los tipos? ¿O para cada prenda será distinto? En la solución implementada asumo que el descuento se define individualmente por cada prenda.

A su vez, a la hora de definir un coeficiente fijo asumo que el coeficiente se aplica por igual a todas las ventas con tarjeta, por lo cual decidí crear un objeto y asignar momentáneamente un valor arbitrario (0.01).

Respecto a decisiones sobre el diseño, el punto en el cual más puede diferir esta solución de otras es la implementación de las ganancias. En este caso decidí crear una clase en la cual cada instancia contendrá una lista de ventas y podrá obtener las ganancias de esa lista en una determinada fecha. La razón de esta decisión es para que el usuario tenga la libertad de crear un objeto que contenga una lista de todas las ventas (lo cual permitiría simplemente crear un objeto directamente en vez de una clase) o distintos objetos para segmentar las ganancias y ventas (por ejemplo, crear un objeto de ganancias por cada mes, cada año, por medio de pago, etc.).

Sin duda no es la única solución y depende de los deseos del cliente pero considero que es la opción que más libertad da al respecto. Otra solución que también satisface los requerimientos iniciales expresados por el cliente sería tener un objeto por cada día y simplemente sumar las ganancias de las ventas que existen en ese objeto.