

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.*

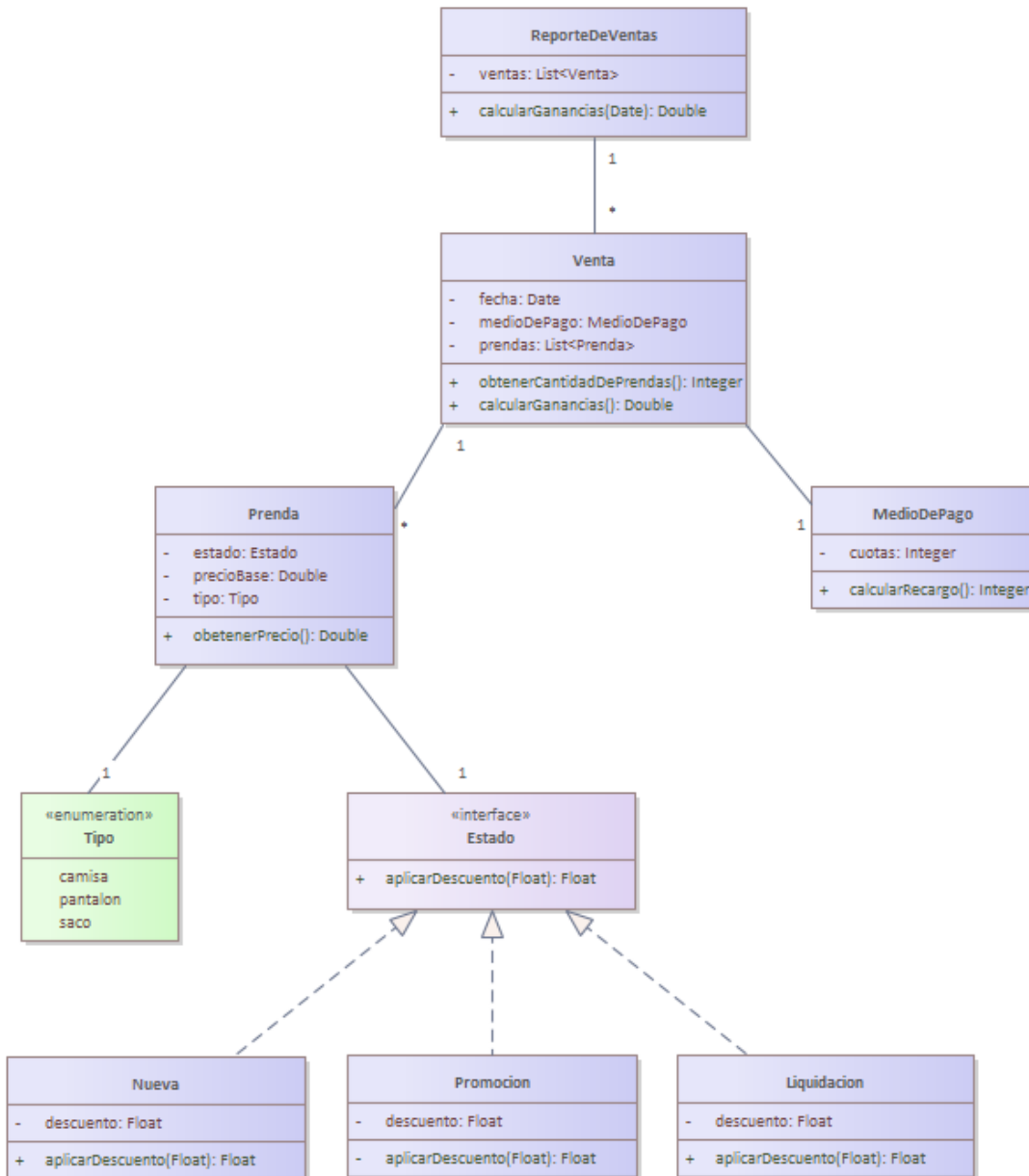
Ah, un requerimiento más: Macowins registra las ventas de estas prendas y necesita saber las ganancias de un determinado día.

*“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).”*

Requerimientos funcionales

- Obtener precio de una prenda
- Obtener tipo de una prenda
- Calcular precio de una prenda según su estado (estados: nueva, promoción, liquidación)
- Obtener ganancias de un determinado día
 - Calcular ganancia de una venta según método de pago (efectivo, tarjeta)

Solución



Aclaraciones

El **Tipo** de una prenda se representa con un *Enum* dado que es de carácter informativo, el comportamiento de una prenda no depende de su atributo **Tipo**.

El **Estado** se representa con una *Interfaz* ya que el descuento se aplica de forma distinta por cada estado (multiplicando por un porcentaje o restándolo), esto se traduce en una implementación diferente del método *aplicarDescuento en cada caso*.

Alternativas descartadas

Descarté utilizar herencia para cada tipo de prenda.

Consideré que no era necesario porque todas las prendas tienen los mismos atributos y comportamiento. Por otro lado, uno de los requerimientos es saber el tipo de prenda, en ese caso tendría que haberle preguntado al objeto de qué clase era, o agregar un atributo “tipo” el cual sería redundante.