

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).”*

Resolución:

Comencé anotando lo que leía y como clasificarlo, es decir, la jerarquía con la relación de los requerimientos y el funcionamiento particular de este.

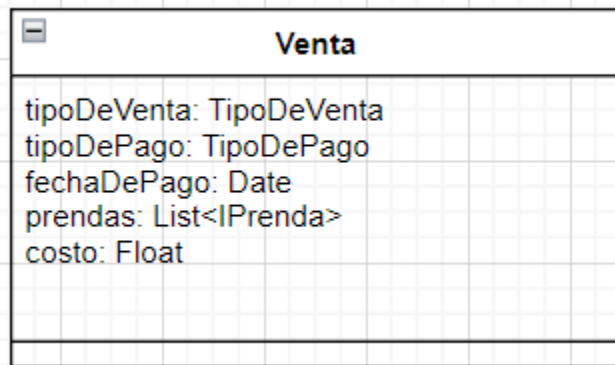
- Precio:
 - Precio propio modificado por
 - Nueva: precio base
 - Promoción: resta valor fijo
 - Liquidación: 50%
- Tipo:
 - Sacos
 - Pantalones
 - Camisas
- Registrar ventas y saber ganancias de un día
- Venta
 - Prendas con su cantidad
 - Fecha de venta
 - Tipo de pago
 - Efectivo
 - precio
 - Tarjeta
 - cuotas * un coeficiente fijo + 0.01 del valor de cada prenda
 - Tipo de Venta
 - Nueva: precio base
 - Promoción: resta valor fijo
 - Liquidación: 50%

Los requerimientos que identifiqué son:

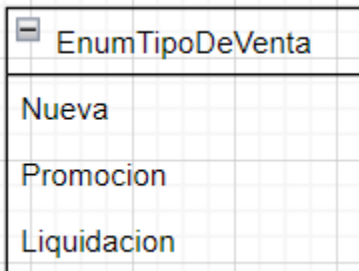
1. Poder calcular el precio de una prenda teniendo en cuenta el estado (esto a pesar de decir estado interpreto que es el tipo de venta, ya que la prenda tenga un estado nuevo y otra promoción no lo veo lógico, ya que la promoción o liquidación no lo interpreto como un estado de prenda (nuevo, viejo, roto).
2. Que se pueda tener un registro de las ventas en donde pueda saber las ganancias del día.

Modelado:

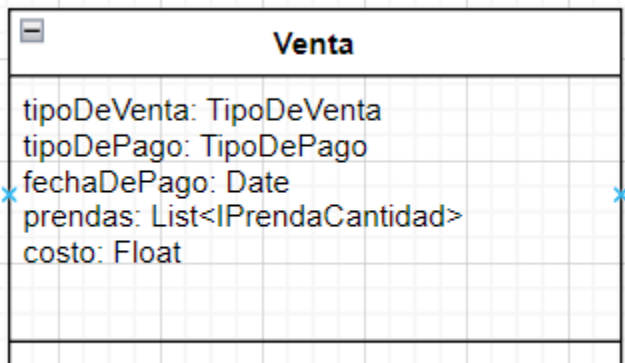
- Para la Venta se me ocurrieron 2 opciones para contemplar el conteo de ventas, primero tener una lista solo de prendas



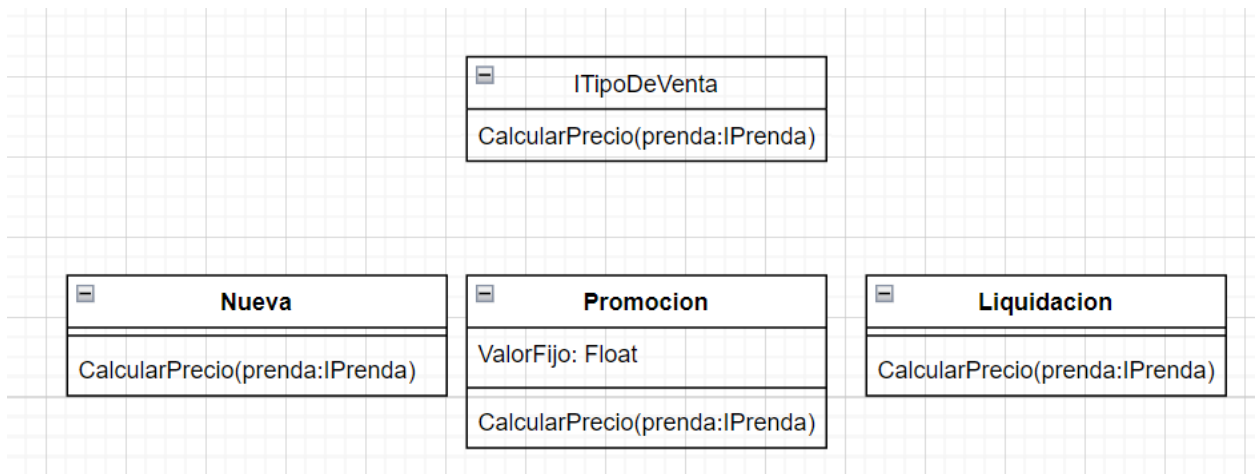
y poder saber la cantidad de prendas que poseen al contar la cantidad de prendas con un `typeOf` o contando post filtro (dependiendo de si la prenda termina siendo una clase o si manejo el tipo de Prenda con un Enum, elijo manejar las prendas por interfaz de `IPrenda`).



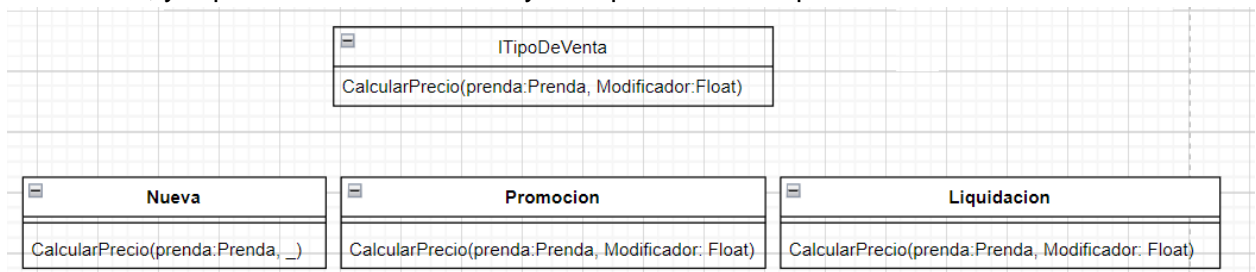
Otra opción que es manejar las prendas por una lista de `prendasCantidad` que posee la prenda y la cantidad comprada de ese tipo de prenda



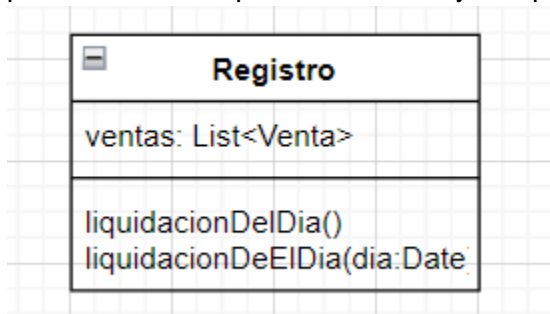
- Para modelar los tipos de prenda (para el sistema tipos de venta) se me ocurrió una implementación de `ItipoDeVenta` que tenga como método el `calcularPrecio` que recibe un `IPrenda` y calcula en base a como funciona cada uno de los tipos, para el de promoción posee aparte un valor fijo que se configura al crear la promoción (internamente también puede haber un archivo de configuración donde se tome ese valor, para no aumentar la complejidad lo manejo por propiedad de entidad, pero prefiero que use el archivo de configuración para mayor flexibilidad).



A su vez otra opción que se me ocurrió es que el método de `calcularPrecio` reciba un modificador con la prenda en la firma, `CalcularPrecio(prenda: Prenda, Modificador: Float)` el cual se usa como corresponda dentro del Tipo de venta específico dentro. En particular esta opción no me gusta, me parece que es forzado el modificador por la **Promocion**, ya que en **nuevo** no se usa y en **liquidación** siempre es 0.5.



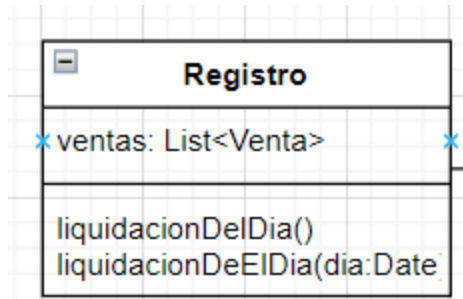
- Decidí tener un **Registro** el cual posee el listado de ventas hechas y posee los métodos para obtener la liquidación del día y la liquidación de un día en específico.



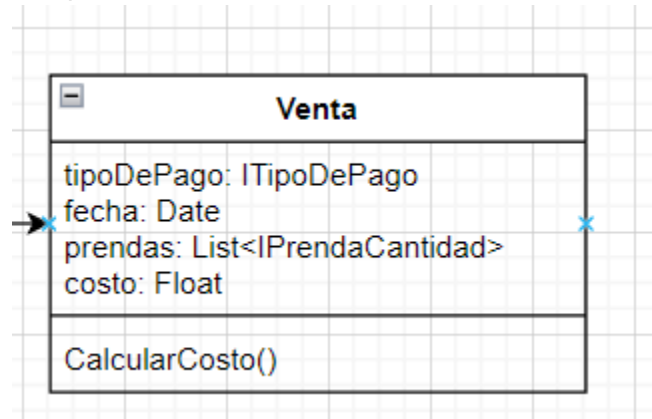
- Repasando la solución me di cuenta de que tengo un error al calcular el precio de la prenda ya que no estaría bien el tener separado el tipo de la prenda (nuevo, promoción, liquidación) con la prenda en sí. Por lo que cambio el modelo

Modelo 2:

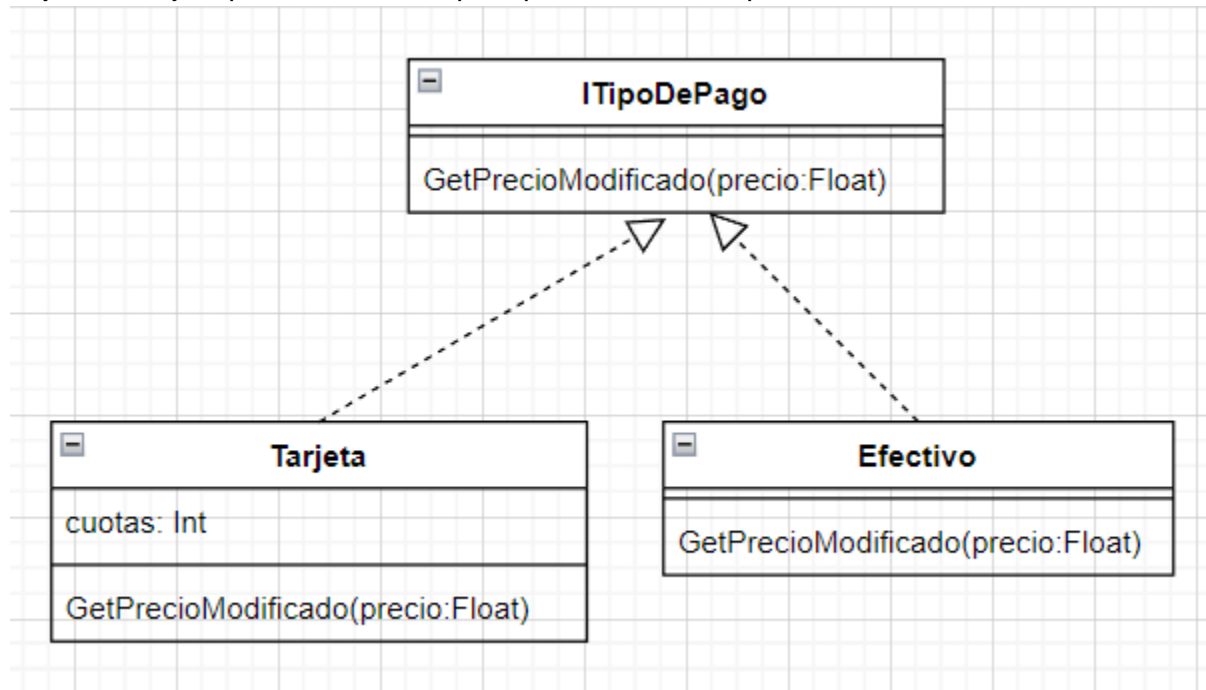
- El registro se mantiene de la misma forma:
 - Posee el listado de ventas hechas y posee los métodos para obtener la liquidación del día y la liquidación de un día en específico.



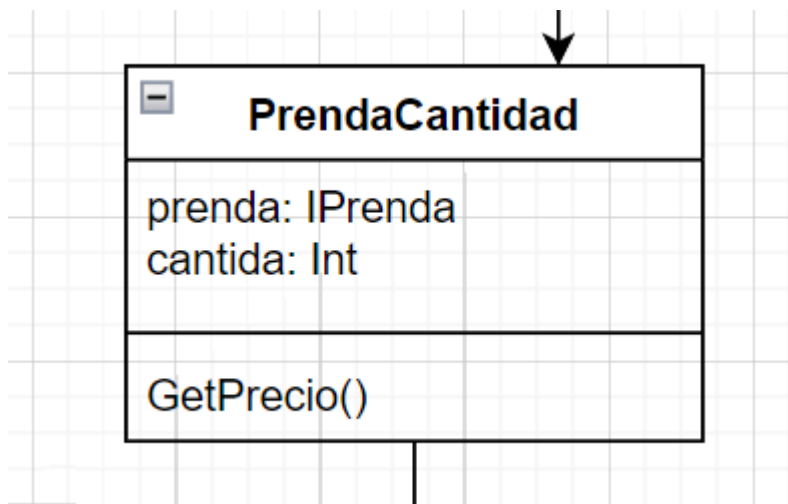
- Para la venta uso una clase Venta que tiene como atributos:
 - `ITipoDePago`: Interfaz que sirve para separar en efectivo o en pago con tarjeta.
 - `fecha`: Fecha en la que se realiza la venta
 - `Prendas`: Lista de `PrendaCantidad`, que posee la prenda y la cantidad de prendas de ese tipo que se compran.
 - `Costo`: Es el costo de la venta, lo dejo como un valor fijo a pesar de poseer el método para calcularlo, ya que puede cambiar el precio de una prenda con el tiempo.



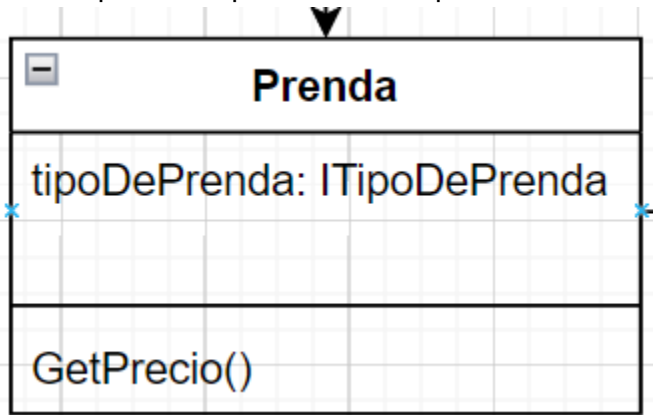
- El ITipoDePago es una interfaz que sirve para poder utilizar el pago en efectivo o la tarjeta, la tarjeta posee las cuotas para poder calcular el precio modificado



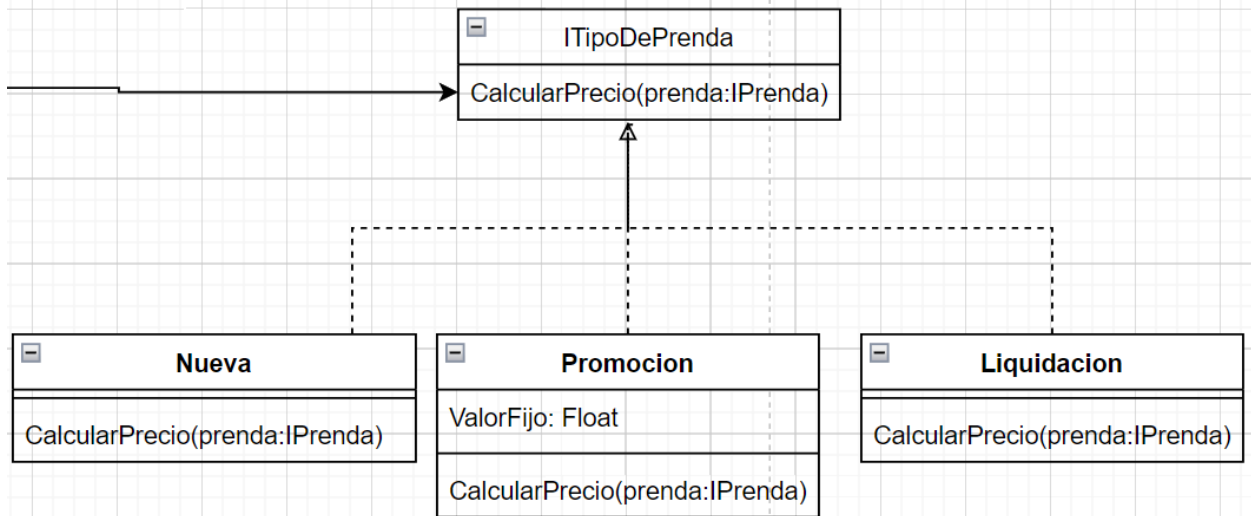
- La entidad de PrendaCantidad posee la prenda y la cantidad de prendas que se hicieron en la venta.



- Prenda posee el tipoDePrenda especifica:



- `ITipoDePrenda` es la interfaz para tratar los tipos de prenda que se necesita y que cada uno implemente la lógica que necesitan.



Modelo final:

