1. Requerimientos:

* Calcular el precio de una prenda según su estado
* Informar el precio de una prenda
* Indicar el tipo de una prenda
* Registrar ventas
* Informar las ganancias de un día indicado

B) Diagrama de Clases

Diagrama

Descripción generada automáticamente

1. Aclaraciones:

Para simular el comportamiento de la empresa Macowins utilicé una clase llamada “Local” que registra las ventas existentes y que también puede consultar las ganancias del total de ventas, según el día indicado. Sin embargo, una de las ideas que no pude concluir es generar un menú donde se va solicitando al usuario, de manera interactiva o “en tiempo real” toda la información relacionada a la venta.

Para el estado de una prenda pensé en una interface que es implementada por tres clases distintas (Nuevo, Promocion, Liquidacion). Utilizando esto considero que puedo seguir agregando distintos estados que puedan surgir en algún futuro y que posiblemente tengan una lógica distinta tanto para calcular los precios como por tener atributos únicos para dichas clases. Pero, por otro lado, aclaro que para lo solicitado actualmente podría haber utilizado una clase y no una interface.

Para la prenda hice una clase abstracta donde sus atributos y algunos métodos son reutilizados en tres clases hijas: Camisa, Saco y Pantalon. En caso de agregarse nuevos tipos de prendas, estaría reutilizando la mayor cantidad de elementos posibles. Considerando esto último y teniendo en cuenta que se pide conocer el tipo de prenda, pensé en diferenciarlos usando clases distintas.

1. **No concluido**: Para registrar las ventas pensé en crear un menú básico (sin interfaz gráfica) donde se solicitaba al usuario del sistema, indicar de forma iterativa la prenda a comprar, su estado para posteriormente agregarlo a una lista de prendas. Luego, instanciar una venta indicando la fecha de la compra y utilizando la lista de prendas generada anteriormente. Una vez instanciada dicha venta, agregarla a una lista de ventas general.

No pude finalizarlo por falta de tiempo y porque la solución que estaba desarrollando estaba hecha de una forma que no era del paradigma de objetos, sino imperativa.