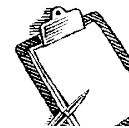


## **Guía de ejercitación nro. 4**



**Tema principal:** Programación orientada a objetos en Smalltalk.

**Temas específicos:** Colecciones. Composición de clases propias. Bloques. Introducción a las colecciones en Smalltalk.



### **Actividades:**

Para el caso de estudio que a continuación se describe, realiza lo siguiente:

- Identifica cada una de las entidades involucradas, sus atributos y responsabilidades. Categoriza las responsabilidades de cada clase en métodos de inicialización, métodos de acceso, métodos modificadores y métodos de negocio según correspondan.
- Establece las relaciones entre las entidades. Realiza el diagrama de clases correspondiente.

### **Caso de estudio:**

Una tienda de ramos generales que comercializa artículos nos solicitó un programa orientado a objetos en Smalltalk que le permita gestionar información respecto a los artículos que comercializa y sus respectivas garantías.

a) De las garantías, se necesitan registrar los siguientes datos:

- teléfono: es una cadena que representa el número de teléfono de contacto de la garantía (valor por defecto: 1111111111).
- duración: es un número entero que representa la cantidad de meses de vigencia de la garantía (valor por defecto: 0).

Las responsabilidades que deberá tener son las básicas, es decir: inicializarse, asignar o modificar un valor a cada uno de sus atributos, acceder al valor de cada uno de sus atributos, acceder a todos los valores juntos de sus atributos, y cualquier otra que se considere conveniente para optimizar la distribución de responsabilidades.

b) De cada artículo se deberán registrar los siguientes datos:

- descripción, es una cadena de caracteres correspondiente al nombre del artículo (valor por defecto: "sin nombre").
- costo, es un valor real que representa el costo de compra en pesos del artículo (valor por defecto: 0.0).

- `porcGanancia`, es un valor real, comprendido entre 0.0 y 200.0, que representa el porcentaje de ganancia sugerido para dicho artículo (valor por defecto: 0.0).
- `stock`, es un valor entero que representa la cantidad de unidades disponibles en stock del artículo (valor por defecto: 0).
- **garantía**, es un objeto de la clase **Garantía**. Un artículo sólo tiene una única garantía.

Las responsabilidades específicas de un artículo, además de las básicas (inicialización, modificación de atributos y acceso a atributos), son las siguientes:

- ✓ Calcular el precio final de venta unitario.
- ✓ Conocer el capital del artículo en términos del stock disponible. Dicho capital se calcula multiplicando el stock disponible del artículo por el precio final de venta de este.

El precio final de venta unitario depende del costo, del porcentaje de ganancia sugerido y de la duración de la garantía.

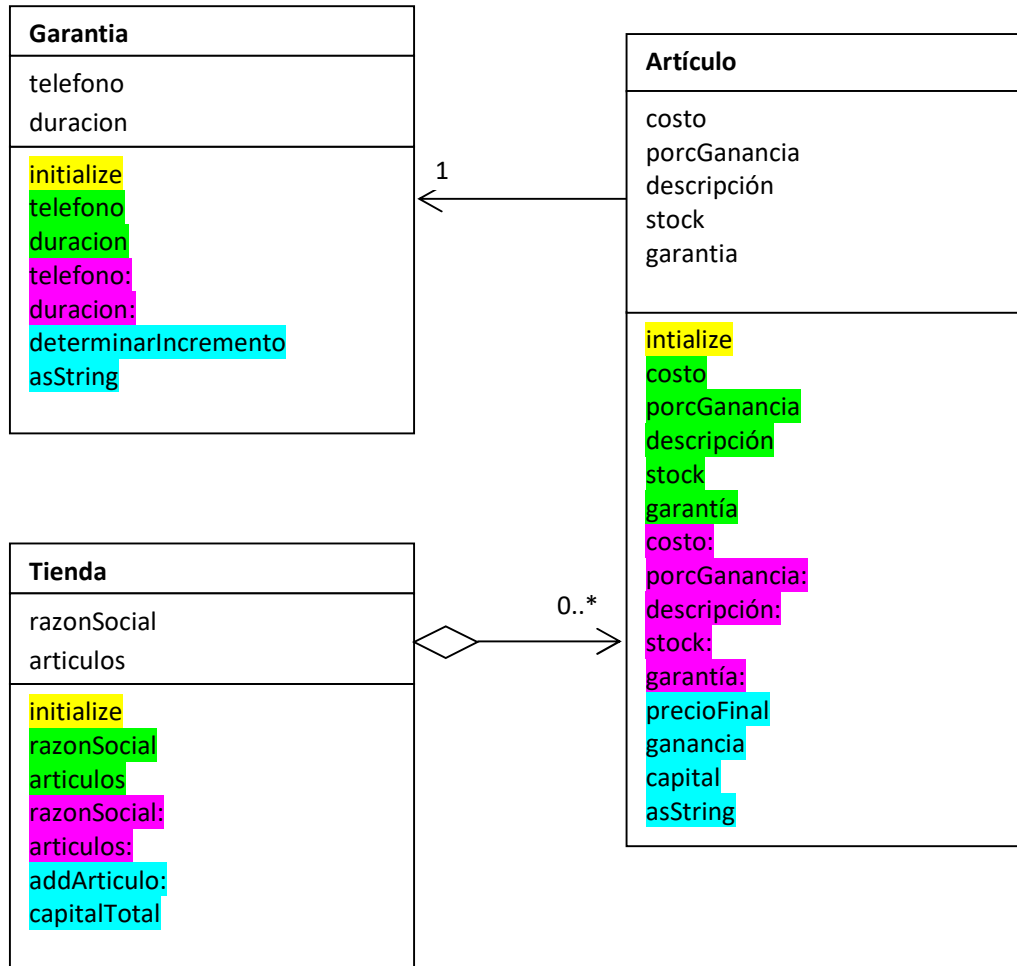
El cálculo del precio final de venta unitario se realiza de la siguiente manera:

- 1) Primero se le suma al costo, la ganancia del artículo, a partir del porcentaje de ganancia sugerido del artículo.
  - 2) Luego, se le incrementa al anterior importe del artículo, un porcentaje de este dependiendo de la duración que tiene la garantía asociada al artículo, según los siguientes criterios:
    - Si la duración es menor a 6 meses, sólo incrementa un 10%.
    - Si la duración es mayor o igual a 6 meses, incrementa un 25%.
- c) De la tienda se deberán registrar los siguientes datos:
- `razónSocial`, es una cadena de caracteres correspondiente al nombre de la tienda (valor por defecto: "anónima").
  - `artículos`: es una colección de todos los artículos que comercializa la tienda; es decir, es una colección compuesta por cero o más objetos de la clase Artículo.

Las responsabilidades específicas de la tienda, además de las básicas (inicialización, modificación de atributos y acceso a atributos), son las siguientes:

- ✓ Agregar un artículo a la tienda.
- ✓ Conocer el capital total que representan todos los artículos que tuviese la tienda.

### Diagrama de clases:



### Actividades de tarea:

Agrega las siguientes responsabilidades para la clase Artículo:

- ✓ Sumar al stock una cierta cantidad de artículos (usar un método binario +=). Se podrá incrementar el stock sólo si la cantidad fuese positiva o 0.
- ✓ Restar del stock una cierta cantidad de artículos (usar un método binario -=). Se podrá decrementar el stock del artículo, siempre y cuando el stock sea mayor o igual a la cantidad recibida como colaborador, y dicho colaborador fuese positivo o 0.
- ✓ Conocer si el stock del artículo es mayor o igual que cierta cantidad (usar el método binario >=)
- ✓ Conocer si el artículo está sin stock.

Agrega las siguientes responsabilidades para la clase Tienda:

- ✓ Conocer el importe final promedio de todos los artículos que comercializa.
- ✓ Conocer si la tienda tiene o no un artículo cuya descripción se especifica como colaborador.
- ✓ Conocer la cantidad de artículos sin stock.