Desafío - Interacciones entre objetos

En este desafío validaremos nuestros conocimientos de abstracción y encapsulamiento, colaboración y composición..

Lee todo el documento antes de comenzar el desarrollo **individual o grupal**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

Te encuentras participando de un proyecto de emprendimiento que consiste en una aplicación móvil de compra y reparto de productos. El equipo ha decidido hacer el backend de la aplicación utilizando Python y el paradigma orientado a objetos. Para el primer prototipo de entrega, se solicita realizar una aplicación de consola en Python, donde los ingresos de valores se hagan mediante input.

El equipo te ha solicitado diseñar e implementar la arquitectura de clases que involucra a la entidad principal "Tienda". Estas son las consideraciones que se deben tener en cuenta respecto de las tiendas:

- Existen por el momento 3 tipos de tienda (en el futuro podría haber más), los cuales son: "Restaurante", "Supermercado" y "Farmacia".

- Todas las tiendas deben poder ***ingresar un producto, listar los productos existentes, y realizar ventas.***

- Cada tienda creada, independiente de su tipo, posee un nombre, un listado de productos y un costo de delivery. Al momento de crear una nueva tienda, se debe solicitar el nombre y el costo de delivery (todas las tiendas se crean inicialmente sin productos). En una tienda ya existente, no se puede modificar el nombre ni el costo de delivery, pero sí se puede modificar los productos (mediante el ***ingreso de un producto***, o mediante la ***realización de ventas***).

- Los productos tienen un nombre, un precio y un stock. Los 3 valores se deben solicitar al momento de crear un producto nuevo, pero si no se indica stock, se asume que es 0. No se puede modificar el nombre ni el precio de un producto, solo su stock. Si se intenta modificar el stock por un valor menor a 0, se debe asignar 0 en su lugar. De cada producto se puede obtener su nombre, su precio o su stock.

NOTA: Se asume que cada producto es específico de una tienda. Es decir, un producto no existe por sí mismo, sino que como parte de una tienda.

Respecto del comportamiento de cada tipo de tienda, considere lo siguiente:

- Para ***ingresar un producto a una tienda,*** se debe solicitar los datos requeridos del producto. Una vez creado el producto, éste se añade a la lista de productos a la tienda. Si el producto ya existe en la tienda (dado por su nombre), se debe modificar su stock, sumando al valor existente el stock del nuevo ingreso. Se conserva el precio del primer ingreso de un mismo producto. ***Tip***: Pruebe sobrecargar los operadores `__add__`, `__sub__` y `__eq__`.

NOTA: Los productos de las tiendas de tipo "Restaurante" siempre tienen stock igual a 0, ya que el producto solo se fabrica al momento de que se realiza una venta. Es decir, aunque se especifique un valor de stock, los productos de estas tiendas se crean con stock 0 y este no se modifica si se añade nuevamente el mismo producto a la lista de productos existentes de la tienda.

- Al ***listar los productos existentes***, se debe ocultar el stock de los productos en el caso de las tiendas de tipo Restaurante y Farmacia. Las tiendas de tipo Supermercado deben añadir el mensaje "Pocos productos disponibles" junto a la cantidad de stock del producto, en caso de que el stock del producto sea inferior a 10. Las tiendas de tipo Farmacia deben añadir el mensaje "Envío gratis al solicitar este producto" junto al precio de los productos con un valor superior a $15.000.

NOTA: Considera que el método para listar los productos será llamado dentro de `print`, por lo que debe retornar un string.

- Para ***realizar una venta,*** se debe solicitar el nombre del producto que se desea vender y la cantidad requerida. Las tiendas de tipo Farmacia y Supermercado deben tener stock existente del producto indicado (si no poseen stock, o no existe el producto solicitado, no se realiza ninguna acción). Sin embargo, los productos de las tiendas de tipo Restaurante siempre tienen stock 0, por lo que no es necesario hacer esta validación ni modificar el stock (***Tip***: puede usar `pass`). Además, en el caso específico de las tiendas de tipo Farmacia, no se puede solicitar una cantidad superior a 3 por producto en cada venta (si se solicita una cantidad mayor a 3, no se realiza ninguna acción). En el caso de las tiendas de tipo Farmacia o Supermercado, si la cantidad

requerida es superior a la existente, solo se venderá la cantidad disponible (quedando entonces el stock del producto en 0).

Requerimientos

1. En un archivo `producto.py`, definir la clase que permita instanciar productos. Considera para la definición de esta clase lo señalado en la descripción de la problemática (utilice ENCAPSULAMIENTO).
   (2 Puntos)

2. En un archivo tienda.py, definir la o las clases necesarias para instanciar los distintos tipos de tienda (utilice ABSTRACCIÓN y ENCAPSULAMIENTO). Cada clase que permita instanciar una tienda debe tener (considerar para cada punto la información entregada en la descripción de la problemática):

   a. Un método constructor **(1 Punto)**
   b. Un método para ingresar un producto (utilice COMPOSICIÓN, y opcionalmente COLABORACIÓN) **(2 Puntos)**
   c. Un método para listar productos **(1 Punto)**
   d. Un método para realizar ventas (utilice COLABORACIÓN) **(2 Puntos)**

Nota 1: *Una clase Compuesta puede tener una lista de Componentes del mismo tipo, almacenados en un atributo de tipo lista.*

Nota 2: *Si utilizas un método sobrecargado, también se considera colaboración.*

3. En un archivo `programa.py`, implementa la lógica necesaria para crear una tienda e ingresar sus productos. Se debe solicitar ingresar productos hasta que el usuario indique lo contrario. Luego, se le debe dar al usuario las opciones de listar los productos existentes, realizar una venta, o salir del programa. Para las primeras dos opciones, debe hacer llamados a los métodos de su instancia y luego volver a consultar cuál de las tres acciones se desea realizar. Si se escoge la tercera opción, se finaliza la ejecución del programa.
   (2 Puntos)

¡Mucho éxito!