

## EXAMEN PARCIAL 1

Leé con cuidado el enunciado y por lo menos dos veces para detectar las clases y estructuras necesarias para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Correcta aplicación de Herencia y Polimorfismo, incluyendo interfaces.
- Correcta implementación de los constructores.
- Correcta validación de los datos que ingresan al sistema.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, utilizando *setters* y *getters* sólo cuando corresponda.
- Generación del Diagrama de Clases.
- Importación y Exportación de proyectos Java desde Eclipse.

***La mala aplicación de los tres primeros puntos hará que el examen sea desaprobado.***

### Enunciado

La nueva Cervecería *Jolybir Artesanal* abrió un local para la venta al público desde donde está dando a conocer sus productos. La empresa produce cerveza de varios sabores predeterminados y sale al mercado para venderla suelta y envasada.

La cerveza suelta se vende en garrafas que se envasan al momento. Los tamaños son GRANDE, MEDIANA y CHICA, de 3 litros, litro y medio y 750 mililitros, respectivamente. La cerveza envasada en fábrica viene en cajones de 12 botellas de 350 mililitros, pero que se venden por unidad.

Los sabores que produce son GOLDEN, BLACK, IPA, STOUT, HONEY, LAGER y BOCK, aunque no todos los días reciben barriles (para las sueltas) y/o cajones de todos los sabores, pues esto varía según la producción.

Para mantener fresca la cerveza el local cuenta con dos refrigeradores, uno para cada clase de Contenedor (Barril y Cajón). Ambos tipos de contenedores son *stockeables*: cuando cualquiera de ellos queda vacío se envía a depósito para saber exactamente qué se debe reponer.

También, a fin de entender el mercado, se lleva la cuenta de cada unidad vendida, sin importar el tamaño pero sí el sabor elegido.

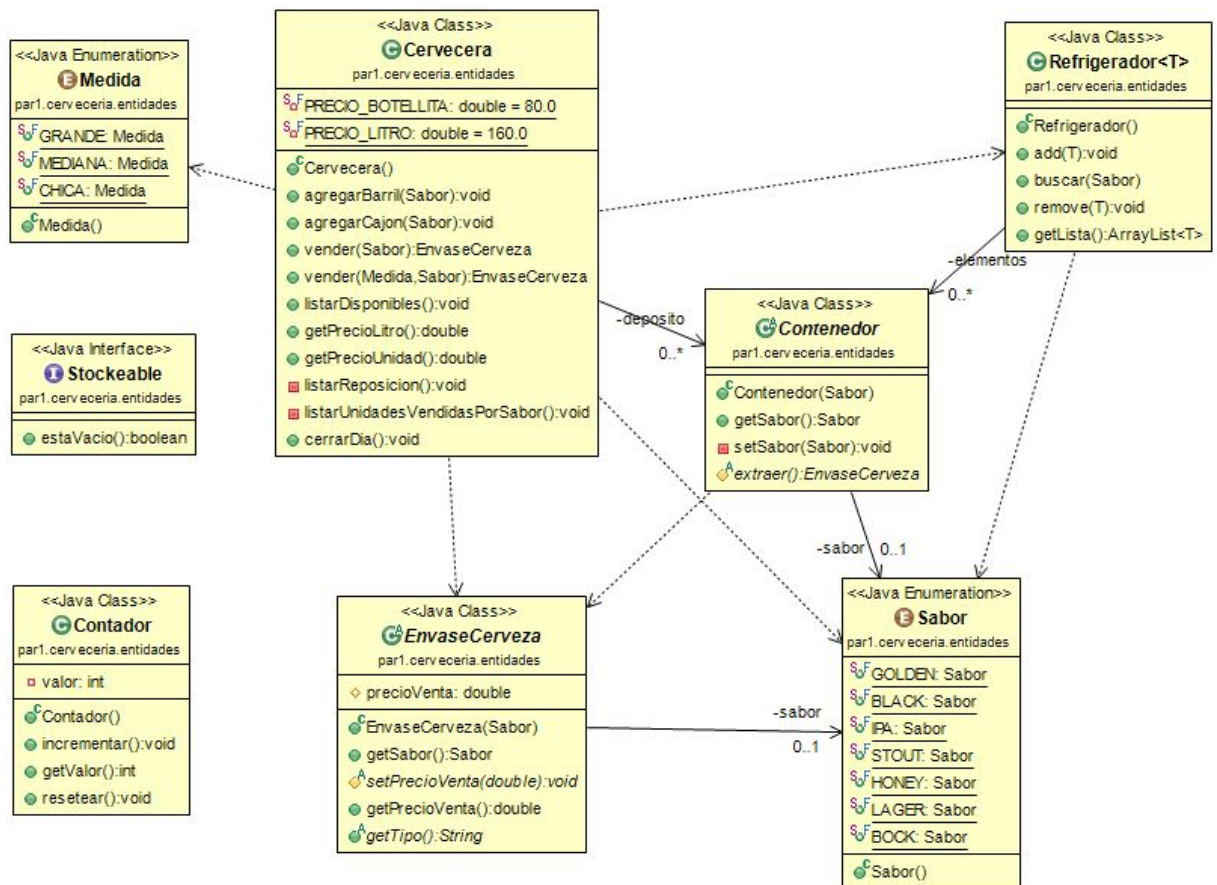
Con cada venta se informa el precio final. Si bien la cerveza suelta tiene como base un precio de 160\$ el litro, el precio final se calcula según el tamaño del envase: a las garrafas medianas se le agrega un 10%, y a las chicas un 20%. Las botellas preenvasadas se venden a 80\$ cada una. Para que el comprador pueda elegir, luego de cargar la mercadería diaria, al abrir el local se listan los sabores disponibles indicando si se trata de cerveza suelta o preenvasada.

Al cerrar la jornada laboral se listan los cajones y barriles vacíos que fueron a depósito y cuántas unidades se vendieron de cada sabor.

**Tomando en cuenta esto, desarrollá los siguientes puntos, siempre demostrando el manejo de los conceptos listados más arriba:**

- a) Descargá del Aula Virtual e importá el proyecto adjunto al anunciado. En él encontrarás una versión incompleta de la solución con el diseño UML que se muestra a continuación:

**EXAMEN PARCIAL 1**



b) Completá el modelo de Clases que represente el enunciado descrito, declarando atributos, métodos y constructores. Plasmalo en un diagrama UML:

1. Implementá las clases **Botella** y **Garrafa**, que extienden **EnvaseCerveza**. A fines prácticos, el tamaño de la garrafa se indica al crearla.
2. Implementá las clases **Cajon** y **Barril**, que extienden **Contenedor**. Se sabe que cada cajón trae 12 botellitas. Se sabe que la capacidad de los barriles es siempre de 30 litros. Los barriles calculan automáticamente la cantidad de cerveza que deben entregar a partir de la medida de la **Garrafa** (grande, mediana o chica). Sabe que la capacidad de la garrafa más grande es de 3 litros y cada una de las siguientes la mitad de la anterior. A pesar de eso, y debido a los hábitos de consumo de los clientes, si al extraer cerveza del barril no se indica la medida de la garrafa, el valor por defecto es MEDIANA.
3. Usá la interfaz **Stockeable** donde te parezca conveniente.
4. Completá la clase **Cervecera**. Asegurate de que al recibir un cajón o un barril se guarde en el refrigerador correcto.
5. La clase **Cervecera** tiene dos métodos **vender()**, uno de ellos sobrecargado. Cada uno está definido para cada tipo de venta, ya que para comprar cerveza preenvasada no hace falta declarar el tamaño del envase. Entonces, uno es para la cerveza suelta y el otro (el primero) es para las botellitas. Además, luego de cada venta controlá si el cajón o el barril usado no está vacío. Si es así debe sacarse del refrigerador y enviarse a depósito, donde se acumulan todos juntos para luego ver la reposición.

c) No hace falta que ingreses datos por teclado: el programa de testeo está completo. Luego de ejecutar el método **main()** de la clase **Ejercicio** el programa debe mostrar lo siguiente:

**EXAMEN PARCIAL 1**

```
El precio por litro es 160.0
El precio por unidad es 80.0
Sabores disponibles de cerveza envasada:
GOLDEN
IPA
LAGER
BOCK
STOUT
HONEY
Sabores disponibles de cerveza suelta:
BOCK
GOLDEN
LAGER
STOUT
** VENTAS **
BOTELLA de STOUT - 80.00$
GARRAFA CHICA de BOCK - 144.00$
GARRAFA MEDIANA de LAGER - 264.00$
No habia cerveza del sabor pedido
GARRAFA GRANDE de GOLDEN - 480.00$
** Cierre del dia **
-- Reposiciones --
No hay nada que reponer.
-- Unidades vendidas por sabor --
GOLDEN = 1
BLACK = 0
IPA = 0
STOUT = 1
HONEY = 0
LAGER = 1
BOCK = 1
```

- d) Cuando termines exportá el proyecto quitándole los binarios y subilos como entrega del examen. No te olvides de actualizar el diagrama UML (debe estar prolijo).