

### **Consignas para la resolución del parcial**



- 1. Importar su proyecto de cine a Eclipse**
- 2. Renombrar el proyecto a “PBII2025Q2P1CineApellidoNombre”**
- 3. Agregar los tests proporcionados.**
- 4. Actualizar el proyecto siguiendo lo descrito en la siguiente página, usando la metodología TDD, utilizando polimorfismo, herencia, y todos los temas del cronograma donde correspondan.**
- 5. Las clases para este proyecto deben encontrarse desacopladas de las del proyecto de cine. Puede separarlo en packages en caso de considerarlo oportuno**
- 6. Entregar el proyecto por prácticas de Miel.**

## Enunciado

### Gestión del Candy Bar de un Cine

2

Se desea extender la funcionalidad de nuestra sala de cine.

El objetivo es crear un sistema para gestionar los productos del candy bar del cine. Deberán modelar los diferentes tipos de productos, calcular los precios y gestionar un inventario simple.

### Requisitos

Se desean crear una lista de Productos que tengan como propiedades y comportamientos comunes a todos los artículos del candy bar. Esta clase debe incluir: Un nombre, un stock, Un precio base, y un método `calcularPrecioFinal()`.

Un producto no existe pero si existen tres clases de producto:

- **Snack**: Representa productos como palomitas de maíz o nachos. El precio final tiene 20% de recargo si es grande y 15% de descuento si es pequeño.
- **Bebida**: Representa refrescos y jugos. El precio final tiene +10% de recargo en concepto de envase si su contenedor es una botella.
- **Combo**: Representa una combinación de un **Snack** y una **Bebida**. El precio final es la suma de los precios de sus componentes, con un **descuento** personalizable

Se desea tener una clase de gestión de inventario `CandyBar` para almacenar objetos de tipo `Producto`.

Esta clase deberá tener métodos para:

- `agregarProducto(Producto)`: Añade un producto al inventario.
- `eliminarProducto(String)`: Elimina un producto por su nombre.
- `obtenerInventario()`: Devuelve el array de productos.

En este caso no se va a representar la venta de cada producto ni el control de stock.

### Condiciones de aprobación.

- Se considerará desaprobado todo proyecto que:
  - a. no compile, o,
  - b. no se identifiquen los atributos necesarios para resolver lo solicitado, o,
  - c. los tipos de datos elegidos para los atributos no sean los adecuados, o,
  - d. no reutilice el código en los casos que sea posible, aplicando los beneficios de la herencia y polimorfismo, o,
  - e. no cumpla con las condiciones mínimas descriptas a continuación.
- Se considerará aprobado, aquel examen que ejecute de manera correcta al menos el 60% basado en la ponderación de los siguientes test
- Se considerará promocionado, aquel examen que ejecute de manera correcta al menos el 80% basado en la ponderación de los siguientes test:

testCalculoPrecioFinalSnack	15%
testCalculoPrecioFinalBebida	15%
testCalculoPrecioFinalCombo	20%
testAgregarProducto	15%
testEliminarProductoExistente	10%
testEliminarProductoNoExistente	10%
testInventarioLleno	15%
	100%