



---

### Aspectos relevantes del examen;

- Duración del examen 2 horas.
- Aprobación del examen con nota mayor o igual a 4 (cuatro).
- Condiciones de aprobación: 60% correcto.
- Implementar los métodos necesarios para realizar las operaciones solicitadas.
- Aplicar los principios de encapsulamiento y buenas prácticas de programación.

### Consigna:

Implementar un sistema de gestión de un comercio mayorista;

### Punto 1: Clases y Herencia

Cree una clase llamada **Producto** que tenga los siguientes atributos: nombre (String), precio (double) y cantidadEnStock (int). Además, implemente un constructor y los métodos getters y setters necesarios.

Cree dos subclases llamadas **ProductoPerecedero** y **ProductoNoPerecedero** que hereden de la clase **Producto**.

**ProductoPerecedero** debe tener un atributo adicional **fechaDeCaducidad** (String), mientras que **ProductoNoPerecedero** debe tener un atributo adicional **tipo** (String). Implemente los constructores y métodos necesarios en ambas subclases.

Implemente un método **calcularPrecioTotal** en ambas subclases que tome un parámetro **cantidadComprada** y calcule el precio total de los productos vendidos teniendo en cuenta su cantidad y precio unitario.

Muestre ejemplos de creación de objetos de ambas subclases, agregándolos a una lista de productos y calcule el precio total en función de una cantidad comprada.

### Punto 2: Interfaces y Polimorfismo

Cree una interfaz llamada **Ordenable** que declare un método **calcularTotal** que tome como parámetro una lista de productos y calcule el precio total de la orden.

Implemente la interfaz **Ordenable** en las clases **ProductoPerecedero** y **ProductoNoPerecedero**. Asegúrese de que los métodos implementados utilicen la información de los atributos de cada producto.

Cree una lista de productos de tipo **Producto** (que incluye tanto **ProductoPerecedero** como **ProductoNoPerecedero**) y calcule el precio total de la orden utilizando polimorfismo y la interfaz **Ordenable**.

### Punto 3: Ordenamiento y Búsqueda

Cree una lista de al menos 10 productos de diferentes tipos (algunos perecederos y otros no perecederos).

Implemente una función que ordene esta lista de productos por nombre de producto en orden alfabético. Puede utilizar el método Collections.sort() o implementar su propio algoritmo de ordenamiento.

Implemente un método que busque un producto por su nombre en la lista y muestre su información si se encuentra. Puede utilizar un enfoque de búsqueda lineal o binaria, dependiendo de su preferencia.

### Punto 4: Diagrama de clases

Grafique el diagrama de clases correspondiente.

Comprobar el correcto funcionamiento en el método main() de la clase **TestComercio**. En dicha clase solo deben existir las instancias a objetos y llamada a métodos, sin resolver el problema aquí.