

PRÁCTICA EVALUABLE

Supongamos que estamos desarrollando un **sistema de gestión de una compañía de logística**. Tendrás que diseñar varias clases para modelar este sistema.

a) Interfaz **Transportable** (1 punto)

Crea una interfaz llamada Transportable con los métodos:

- **calcularCostoEnvio**(boolean esInternacional) que devuelve el costo de envío del objeto que implementa la interfaz.
- boolean **esEnvioInternacional**()

b) Clase **Producto** (2 puntos)

Crea una clase Producto con los siguientes atributos:

- nombre (String)
- peso (double)

Incluye un constructor y métodos getter y setter para cada atributo. Implementa la interfaz Transportable en esta clase: el costo base es de 20€, si es internacional le sumamos 30€. Le sumamos un coste por peso (peso * 0.2).

c) Subclases de Producto (2 puntos)

Crea dos subclases de Producto:

- **ProductoFragil**: Agrega un atributo nivelFragil que indica si el producto es frágil (1) o muy frágil (2). Para el cálculo del coste del envío le suma 20€ si es frágil y 50€ si es muy frágil.
- **ProductoRefrigerado**: Agrega un atributo nivelRefrigeracion que indica si el producto requiere refrigeración isotérmica (1) o refrigeración frigorífica (2). El primero añade un coste de 10€ al envío y el segundo de 30€.

d) Clase **Envio** (2 puntos)

Crea una clase Envio que contenga un ArrayList de objetos que implementen la interfaz Transportable. Además, lleva dos campos, fechaSalida y fechaLlegada. El constructor no lleva ningún parámetro, la fecha de salida es ahora, la de llegada es null, y hay que inicializar el ArrayList. Además de getters y setters necesarios, Incluye los siguientes métodos:

- **agregarTransportable**(Transportable item): Agrega un objeto transportable al envío.
- **calcularCostoTotalEnvio**() : Calcula y devuelve el costo total de envío de todos los objetos en el envío.

- **listarProductos()**: Muestra la información de todos los productos en el envío, con el coste de cada uno.
- **diasTranscurridos()**: devuelve los días que han pasado entre la fecha de salida del almacén y la de llegada a destino.

e) Clase **ServicioEnvio** (2 puntos)

Crea una clase ServicioEnvio que contenga un ArrayList de envíos. Incluye los siguientes métodos:

- **agregarEnvio**(Envio envio): Agrega un envío a la lista de envíos del servicio.
- **generarInformeEnvios()**: Genera un informe que incluye la lista de envíos y su costo total. Debe mostrar todos los productos del envío, el coste total del envío, y los días transcurridos entre la fecha de salida y de llegada.

f) Escribe un programa principal **Main** (1 punto) que:

- Cree al menos seis productos, tres de cada subclase.
- Cree un envío y agregue tres productos de diferente tipo al envío. Asigne fecha de llegada.
- Crea otro envío y agregue los otros tres productos a este envío. Asigne fecha de llegada.
- Agregue los envíos a un servicio de envío.
- Genere e imprima el informe de envíos.