

Ejercicio Almacén Genérico Parte 3



Enunciado: Almacén Genérico con Expresiones Lambda e Interfaces Funcionales

En esta tercera parte, vamos a extender la funcionalidad del sistema para permitir operaciones más avanzadas usando interfaces funcionales y expresiones lambda.

Requisitos:

1. Filtrado de Elementos:

- Agrega un método `filtrar(Predicate<? super T> criterio): List<T>` en la interfaz `Almacenable<T>`. Este método debe permitir obtener una lista de elementos que cumplan con un criterio dado.
- Implementa este método en la clase `Almacen<T>` para que devuelva una lista de elementos que satisfacen la condición del `Predicate` proporcionado.

2. Aplicación de Acciones a Elementos:

- Agrega un método `paraCadaElemento(Consumer<? super T> accion)` en `Almacenable<T>`. Este método deberá aplicar la acción especificada a cada elemento almacenado.
- Implementa este método en `Almacen<T>`, permitiendo recorrer todos los elementos y aplicar la acción provista mediante una expresión lambda.

3. Transformación de Elementos:

- Agrega un método `transformar(Function<? super T, ? extends T> transformacion): List<T>` en la clase `Almacen<T>`. Este método debería aplicar una función de transformación a cada elemento del almacén y devolver una nueva lista con los resultados.
- Usa el método en la clase `PruebaAlmacen` para demostrar cómo transformar los elementos, por ejemplo, aumentando el precio de todos los productos de tipo "comestible" en un 10%.

4. Clase de Prueba:

- En `PruebaAlmacen`, demuestra el uso de estas nuevas funcionalidades:
 - Filtra productos que sean de tipo "electrónico".
 - Aplica una acción para imprimir todos los productos que tengan un precio superior a un cierto valor.
 - Transforma todos los productos para actualizar su precio o cambiar sus atributos según sea necesario.