

## Ejercicio Interface Genérica Parte 2

### Enunciado: Almacén Genérico con Ordenamiento e Iteración

En esta segunda parte del ejercicio, se espera que amplíes la funcionalidad del sistema de almacenamiento genérico que implementaste en la primera parte. Ahora, deberás agregar la capacidad de ordenar los elementos almacenados y permitir su recorrido de manera flexible tanto en su orden natural como en un orden personalizado mediante el uso de un Comparator.

#### Requisitos:

##### 1. Interfaz Almacen<T>:

- Debes mantener la misma interfaz genérica **Almacenable<T>** que definiste en la primera parte, con los métodos:
  - **void agregar(T item)**
  - **T obtener(int indice)**
  - **void eliminar(int indice)**
  - **void eliminarPorIndice(int indice)**
  - **boolean contiene(T item)**
  - **int tamaño()**

##### 2. Clase Almacen<T>:

- Implementa la interfaz **Almacenable<T>** utilizando una lista interna para almacenar los elementos.
- La clase debe implementar la interfaz **Iterable<T>** para que pueda ser recorrida utilizando un bucle **for-each**.
- Implementa un método **iterator()** que devuelva un iterador que recorra los elementos en su orden natural (si el tipo **T** lo permite, es decir, si implementa Comparable).
- Implementa una sobrecarga del método **iterator(Comparator<? super T> comparator)** que devuelva un iterador que recorra los elementos en el orden especificado por el Comparator recibido como parámetro.

##### 3. Métodos para mostrar contenido:

- Implementa un método **mostrarContenido()** que utilice el iterador predeterminado (orden natural) para mostrar los elementos del almacén.
- Implementa otro método **mostrarContenido(Comparator<? super T> comparator)** que utilice el iterador con el Comparator para mostrar los elementos en un orden personalizado.

##### 4. Clase Producto:

- Crea una clase Producto que implemente **Comparable<Producto>**, donde los productos se ordenen por su atributo id (de menor a mayor).

- La clase debe incluir un método **toString()** para imprimir los productos de manera legible, mostrando su id y nombre.

## 5. Clase de Prueba:

- Crea una clase de prueba que:
  - Cree un almacén de productos (**AlmacenLista<Producto>**).
  - Agregue al menos 3 productos con diferentes id y nombre.
  - Muestre los elementos utilizando el método mostrarContenido() en **orden natural** (por id).
  - Muestre los elementos utilizando el método mostrarContenido() con un **Comparator** que ordene los productos por nombre.
  - Demuestre que el almacén es **iterable** utilizando un bucle **for-each** para recorrer los elementos tanto en el **orden natural** como con el **Comparator de nombre**.

### Extra :

- Puedes implementar métodos adicionales o mejoras, como permitir ordenar por otros atributos en la clase Producto, o agregar validaciones para manejar casos como índices fuera de rango.