

Ejercicio Interface Genérica Parte 1

Enunciado: Sistema de Almacenamiento Genérico

Se desea desarrollar un sistema de almacenamiento genérico en Java que permita guardar y recuperar diferentes tipos de objetos de una colección. Para ello, se debe crear una interfaz genérica **Almacenable<T>** que defina los métodos necesarios para agregar, obtener y eliminar elementos.

1. Interfaz **Almacenable<T>**:

Define los siguientes métodos:

- **void agregar(T item):** agrega un elemento al almacén.
- **T obtener(int indice):** obtiene un elemento según su posición en el almacén.
- **void eliminarPorIndice(int indice):** elimina un elemento según su posición en el almacén.
- **boolean eliminar(T item);**
- **boolean contiene(T item):** indica si un elemento esta almacenado.
- **int tamaño():** devuelve la cantidad de elementos almacenados.

2. Implementación **Almacen<T>**:

Implementa la interfaz **Almacenable<T>** utilizando una lista (**ArrayList** o **LinkedList**) como estructura interna de datos.

3. Clase de Prueba:

- Crea una clase de prueba llamada **PruebaAlmacen**.
- Dentro de esta clase, crea un almacén para diferentes tipos de objetos, como **String**, **Integer**, y una clase personalizada (por ejemplo, **Producto**).
- Demuestra la funcionalidad del sistema agregando, obteniendo y eliminando elementos de cada almacén.

4. Extra:

- Agrega una clase **Producto** con atributos como **id** , **nombre**, **precio** y un **enum Tipo**.
- Agrega un método **mostrarContenido()** que recorra cualquier tipo de **Almacenable<T>** e imprima cada elemento.(Utilizar un **Iterador**)

Ejemplo de salida esperada:

```
Almacén de String:
Agregar: "Manzana"
Agregar: "Banana"
Contenido del Almacén: ["Manzana", "Banana"]
```

```
Almacén de Integer:
Agregar: 10
Agregar: 20
Eliminar índice 0
Contenido del Almacén: [20]
```

```
Almacén de Producto:
Agregar: Producto(id=1, nombre="Lapiz")
Agregar: Producto(id=2, nombre="Cuaderno")
```