

Nombre: [titomartir/Pilas-y-Colas: Ejercicios de clases](https://github.com/titomartir/Pilas-y-Colas-Ejercicios-de-clases)

## Inversión de una lista utilizando una pila

### 1. Descripción del TDA:

Este problema utiliza una pila para invertir una lista de elementos. Debido al protocolo LIFO (Last In, First Out) de las pilas, los elementos que se insertan en un orden específico se extraen en el orden inverso. Esto permite invertir una lista o arreglo de manera eficiente.

### 2. Invariante de TDA:

- La pila se utiliza como almacenamiento temporal para los elementos de la lista.
- El orden de los elementos en la lista original se invierte al pasarlos a la pila y luego recuperarlos.
- La lista original se sobrescribe con los elementos en orden inverso.

### 3. Operaciones:

Nombre de operación: invertirLista

- Descripción: Invierte el orden de los elementos en una lista utilizando una pila.
- Descripción operacional:
  1. Se crea una pila vacía.
  2. Se recorren los elementos de la lista y se insertan en la pila.
  3. Se extraen los elementos de la pila y se sobrescriben en la lista original.
- Precondición: La lista no debe estar vacía.
- Postcondición: La lista original contiene los elementos en orden inverso.

### Código con Precondiciones y Postcondiciones

java

Copy

```
import java.util.Stack;
```

```
public class InversionLista {
```

```
    public static void invertirLista(String[] lista) {
```

```
        // Precondición: La lista no debe estar vacía.
```

```
        if (lista.length == 0) {
```

```
            System.out.println("La lista está vacía. No hay elementos para invertir.");
```

```
        return;
    }

    // Crear una pila vacía
    Stack<String> pila = new Stack<>();

    // Insertar todos los elementos de la lista en la pila
    for (String elemento : lista) {
        pila.push(elemento); // Insertar en la pila
    }

    // Extraer los elementos de la pila y sobrescribir la lista original
    for (int i = 0; i < lista.length; i++) {
        lista[i] = pila.pop(); // Extraer de la pila y asignar a la lista
    }

    // Postcondición: La lista original contiene los elementos en orden inverso.
}

public static void main(String[] args) {
    // Lista de ejemplo
    String[] lista = {"A", "B", "C", "D", "E"};

    System.out.println("Lista original: " + String.join(", ", lista));

    // Invertir la lista
    invertirLista(lista);

    System.out.println("Lista invertida: " + String.join(", ", lista));
}
}
```

## Explicación del código

### 1. Estructura de datos:

- Se utiliza una pila (Stack) para almacenar temporalmente los elementos de la lista.

### 2. Proceso de inversión:

- Paso 1: Se recorre la lista y se insertan todos los elementos en la pila. Como la pila sigue el protocolo LIFO, el último elemento de la lista será el primero en la pila.
- Paso 2: Se extraen los elementos de la pila y se sobrescriben en la lista original. Esto invierte el orden de los elementos.

### 3. Salida:

- Se muestra la lista original y la lista invertida.

## Ejemplo de ejecución

### Entrada:

La lista original es: ["A", "B", "C", "D", "E"].

### Salida:

#### Copy

Lista original: A, B, C, D, E

Lista invertida: E, D, C, B, A

## Resumen

Este programa demuestra cómo utilizar una pila para invertir una lista de elementos. La pila actúa como una estructura auxiliar que permite invertir el orden de los elementos de manera eficiente. Es una aplicación clásica del TDA Pila y su protocolo LIFO.