REPORTE DE EJECUCIÓN – ACTIVIDAD 2

Estructuras Dinámicas: Pila y Cola con Listas Enlazadas

Alumno: Rodrigo Mejía
Carrera: Ingeniería en Desarrollo de Software
Institución: Universidad Tecmilenio Las Torres
Fecha: Noviembre 2 del 2025

Introducción

En esta práctica se implementaron las estructuras de datos **Pila** y **Cola**, utilizando una **Lista Enlazada Genérica** como base principal.
El objetivo fue reforzar el uso de **clases genéricas, encapsulación, reutilización de código y estructuras dinámicas** en Java.
A diferencia de la Actividad 1, donde solo se manejaba una lista enlazada, ahora se aplicó esa lógica para crear estructuras específicas (pila y cola) reutilizando la misma clase base.

Objetivos

- Implementar estructuras de tipo **Pila** y **Cola** mediante el uso de listas enlazadas.
- Aplicar el concepto de **encapsulación** en todas las clases.
- Reutilizar la clase LinkedList para crear nuevas estructuras.
- Validar el funcionamiento del código mediante pruebas en la clase Main.

Desarrollo del programa

Clases utilizadas

1. **Node**
   Clase genérica que representa un nodo de la lista enlazada. Contiene los atributos data, next y prev.
2. **LinkedList**
   Clase encargada de manejar la lista enlazada. Puede configurarse como

simple, **doblemente enlazada** o **circular** mediante los parámetros del constructor.

3. **Pila**
   Implementa una pila con operaciones push(), pop() y peek(). Reutiliza la clase LinkedList.

4. **Cola**
   Implementa una cola con operaciones enqueue(), dequeue() y peek(). También reutiliza LinkedList.

5. **Main**
   Clase principal donde se crean objetos de tipo Pila y Cola para probar su funcionalidad.

Principales cambios respecto a la Actividad 1

Elemento	Situación anterior	Situación actual
Clases usadas	Se usaban 4 clases: Node, LinkedList, Main y una clase de prueba adicional.	Se usan 5 clases principales: Node, LinkedList, Pila, Cola y Main.
Estructura	Solo se manejaba una lista enlazada.	Ahora se crearon estructuras derivadas (pila y cola) usando la lista base.
Encapsulación	Algunos atributos eran públicos o se accedían directamente.	Todos los atributos son privados y se acceden mediante métodos.
Uso de genéricos	Limitado a la lista enlazada.	Ahora todas las clases usan <T> correctamente.
Reutilización de código	Cada estructura se implementaba por separado.	Pila y Cola usan directamente la misma LinkedList.
Reflexión / Reflection API	Se usaba para manipular el campo head.	Se eliminó para mantener encapsulación segura.

Ejecución del programa

Descripción de la ejecución

Al ejecutar la clase Main, se realizan las siguientes acciones:

1. Se crea una pila de tipo Integer y se insertan valores con push().
2. Se imprime su contenido, se elimina el último elemento con pop() y se muestra nuevamente.
3. Se crea una cola de tipo String, se insertan elementos con enqueue(), se imprime y luego se elimina el primero con dequeue().

Resultados en consola

```
run:
=== DEMOSTRACION DE PILA Y COLA ===

>>> PILA <<<
Elementos: C B A
Elemento desapilado: C
Tope actual: B
Elementos: B A

>>> COLA <<<
Elementos: 1 2 3
Elemento desencolado: 1
Frente actual: 2
Elementos: 2 3

=== FIN DE LA DEMOSTRACION ===
BUILD SUCCESSFUL (total time: 0 seconds)
```

Conclusión

En esta práctica se reforzaron los conceptos de **estructuras dinámicas, reutilización de código y genéricos** en Java.
La implementación de Pila y Cola demuestra cómo una sola clase (LinkedList) puede servir como base para múltiples estructuras de datos.
Además, se aplicó correctamente la **encapsulación** y el **principio de abstracción**, evitando accesos directos a los atributos y manteniendo un diseño limpio y escalable.