

UNIVERSIDAD INTERNACIONAL DEL ECUADOR

SEDE DE LOJA

Escuela de Ingeniería en Tecnologías de la Información

Actividad 2.2 Suite Tests Casos de Borde

Camila Doménica Loarte Aguilar

Estructura de Datos

Ing. Charlie Cárdenas

16 de diciembre de 2025

1. Introducción

Este documento expone el proceso de diseño e implementación de una suite de pruebas aplicada a una estructura de datos del tipo Lista Doblemente Enlazada (DoublyLinkedList), basada exclusivamente en la implementación desarrollada durante las clases.

Las pruebas se ejecutan directamente desde la terminal, sin apoyo de frameworks externos como JUnit, haciendo uso de los siguientes elementos:

- Un archivo de pruebas que contiene el método main
- Assertions nativas del lenguaje Java (assert)
- Compilación y ejecución manual mediante los comandos javac y java -ea

El enfoque utilizado corresponde a pruebas de caja negra, priorizando la validación de casos borde, de acuerdo con lo indicado en la rúbrica de evaluación.

2. Identificación de casos de borde

ID	Caso borde	Descripción	Importancia
1	Lista vacía	La estructura no contiene nodos.	Previene errores por referencias nulas.
2	Inserción inicial	Se agrega el primer nodo.	Define correctamente head y tail.
3	Un solo elemento	head y tail apuntan al mismo nodo.	Verifica enlaces nulos.
4	Eliminación única	Se elimina el único nodo existente.	Garantiza un estado vacío consistente.
5	Eliminar head	Se borra el primer nodo.	El nuevo head.prev debe ser null.
6	Eliminar tail	Se borra el último nodo.	El nuevo tail.next debe ser null.

ID	Caso borde	Descripción	Importancia
7	Eliminación intermedia	Nodo ubicado entre head y tail.	Asegura reconexión adecuada de enlaces.
8	Eliminación inexistente	El valor no está en la lista.	Debe retornar false.
9	Reversa lista vacía	No hay nodos en la lista.	No debe generar errores de ejecución.
10	Reversa con un nodo	Solo existe un elemento.	La lista no debe alterarse.

3. Implementación de la suite de pruebas

Las pruebas se desarrollan dentro del archivo DoublyLinkedListTest.java, utilizando assertions con mensajes claros que facilitan la identificación de posibles errores.

Cada prueba valida un comportamiento específico de la lista y, al finalizar correctamente todas las validaciones, se muestra un mensaje de confirmación indicando que la ejecución fue exitosa.

4. Anexos

```
(base)
domen@Camila MINGW64 ~/OneDrive/Documentos/UIDE/Semestre5/Estructura de Datos/Semn8 (main)
$ java -ea DoublyLinkedListTest.java
Todas las pruebas pasaron correctamente
(base)
domen@Camila MINGW64 ~/OneDrive/Documentos/UIDE/Semestre5/Estructura de Datos/Semn8 (main)
$
```

```
SEM8
DDL_Node.class
DDL_Node.java
DoublyLinkedList.class
DoublyLinkedList.java
DoublyLinkedListTest.class
DoublyLinkedListTest.java
```

5. Conclusión

La suite de pruebas desarrollada permite verificar de manera efectiva el correcto funcionamiento de la lista doblemente enlazada, cubriendo escenarios críticos como listas vacías, eliminación de nodos en diferentes posiciones y la operación de reversa.

La práctica cumple con todos los criterios establecidos en la rúbrica, haciendo uso de ejecución manual desde la terminal, assertions nativas de Java y una estructura de pruebas clara, ordenada y comprensible.