

ESTRUCTURA DE DATOS
ING. TECNOLOGÍAS DE LA INFORMACIÓN

UNIVERSIDAD INTERNACIONAL DEL ECUADOR

ALUMNO:
Anthony Iván Sánchez Granda

CICLO 5 - PARALELO A

SEMESTRE:
Octubre - Febrero 2026

DOCENTE:
ING. Charlie Cardenas Toledo

ESTRUCTURA DE DATOS
ING. TECNOLOGÍAS DE LA INFORMACIÓN

TA-2.2 — Suite de tests (casos borde)

PARTE 1: Identificación de Casos Borde

Número	Caso Borde	Descripción	¿Por qué es crítico?
1	Lista vacía	head == null	Evita NullPointerException al operar sobre lista sin nodos
2	Un solo elemento	head == tail	Verifica correcto manejo de punteros prev y next
3	Insertar una lista vacía	Insertar primer nodo	Confirma inicialización correcta de head y tail
4	Eliminar head	Se elimina el primer nodo	head.prev debe quedar en null
5	Eliminar tail	Se elimina el último nodo	tail.next debe quedar en null
6	Eliminar elemento intermedio	Nodo con prev y next	Verificar conexión correcta de punteros
7	Eliminar elemento inexistente	Valor no presente	La operación no debe modificar la lista
8	Invertir lista vacía	Lista sin nodos	No debe generar errores
9	Invertir lista con 1 elemento	Un solo nodo	La lista debe permanecer igual

ESTRUCTURA DE DATOS
ING. TECNOLOGÍAS DE LA INFORMACIÓN

PARTE 2: Diseño de Casos de Prueba

ID	Precondición	Acción	Resultado Esperado	Postcondición
TC-DLL-001	Lista vacía	InsertAttEnd(10)	Lista no vacía	head != null
TC-DLL-002	Lista vacía	deleteByValue(10)	Retorna false	head == null
TC-DLL-003	Lista con 1 elemento	deleteByValue(10)	Retorna true	head == null, tail == null
TC-DLL-004	Lista [10,20]	deleteByValue(10)	Elimina head	head.prev == null
TC-DLL-005	Lista [10,20]	deleteByValue(20)	Elimina tail	tail.next == null
TC-DLL-006	Lista [10,20,30]	deleteByValue(20)	Elimina intermedio	Enlaces correctos
TC-DLL-007	Lista [10]	deleteByValue(99)	Retorna false	Lista intacta
TC-DLL-008	Lista vacía	reverse()	Sin errores	Sigue vacía
TC-DLL-009	Lista [10]	reverse()	Sin cambios	Un solo nodo
TC-DLL-010	Lista [1,2,3]	reverse()	Lista invertida	Enlaces correctos

PARTE 3: Implementación de Pruebas

```
==== EJECUCIÓN DE PRUEBAS DOUBLY LINKED LIST ====
?
? TC-DLL-001 Insertar en lista vacía ? OK
? TC-DLL-002 Eliminar en lista vacía ? OK
? TC-DLL-003 Eliminar único elemento ? OK
? TC-DLL-004 Eliminar head ? OK
? TC-DLL-005 Eliminar tail ? OK
? TC-DLL-006 Eliminar elemento intermedio ? OK
? TC-DLL-007 Eliminar elemento inexistente ? OK
? TC-DLL-008 Invertir lista vacía ? OK
? TC-DLL-009 Invertir lista con un elemento ? OK
? TC-DLL-010 Invertir lista con varios elementos ? OK

? TODAS LAS PRUEBAS SE EJECUTARON CORRECTAMENTE
```

ESTRUCTURA DE DATOS
ING. TECNOLOGÍAS DE LA INFORMACIÓN

Parte 4: Reporte de Cobertura

Métrica	Cantidad
Total de pruebas	10
Pruebas ejecutadas	10
Pruebas aprobadas	10
Pruebas fallidas	0

Evidencia de Ejecución

```
==== EJECUCIÓN DE PRUEBAS DOUBLY LINKED LIST ====

? TC-DLL-001 Insertar en lista vacía ? OK
? TC-DLL-002 Eliminar en lista vacía ? OK
? TC-DLL-003 Eliminar único elemento ? OK
? TC-DLL-004 Eliminar head ? OK
? TC-DLL-005 Eliminar tail ? OK
? TC-DLL-006 Eliminar elemento intermedio ? OK
? TC-DLL-007 Eliminar elemento inexistente ? OK
? TC-DLL-008 Invertir lista vacía ? OK
? TC-DLL-009 Invertir lista con un elemento ? OK
? TC-DLL-010 Invertir lista con varios elementos ? OK

? TODAS LAS PRUEBAS SE EJECUTARON CORRECTAMENTE
```

Operaciones Cubiertas

Operación	Cobertura
Inserción en lista vacía	SI
Eliminación en lista vacía	SI
Eliminación del head	SI
Eliminación del tail	SI
Eliminación de elemento medio	SI

ESTRUCTURA DE DATOS
ING. TECNOLOGÍAS DE LA INFORMACIÓN

Eliminación del único nodo	SI
Búsqueda de elemento inexistente	SI
Inversión de lista vacía	SI
Inversión de lista con 1 nodo	SI
Inversión de lista con varios nodos	SI

Identificación de Gaps:

Gap identificado	Descripción
Verificación directa de punteros	No se valida explícitamente head.prev == null o tail.next == null
Inserciones múltiples consecutivas	No se prueba un alto número de inserciones
Comportamiento de display()	No se valida la salida visual del método
Eliminación repetida	No se prueba eliminar el mismo valor dos veces
Estrés de la estructura	No se prueba con listas grandes (100+ nodos)

Propuestas de Mejora para Aumentar la Cobertura

- 1.Exponer métodos getters para head y tail y validar punteros directamente.
- 2.Implementar pruebas de estrés con grandes volúmenes de datos.
- 3.Incluir pruebas de integridad estructural tras múltiples operaciones combinadas.