

**Act 2.2**

Elizabeth Jauregui Zarate

A01253381

El código NodoLista.cpp implementa una lista enlazada simple que utiliza el tipo de datos abstractos (ADT). Este tipo de estructura permite realizar diversas operaciones conocidas como CRUD (Create, Read (buscar), Update, Delete). El objetivo de la utilización de este tipo de datos es el manejo eficiente y amplio de información, como en este caso es listas de elementos de tipo int.

Para validar la correcta validación de entradas y funcionamiento del código, se deben realizar múltiples casos de prueba. Estos casos de prueba deben contener distintos elementos para poder identificar aquellas áreas de errores que a primera vista no se puedan identificar. De esta manera, los casos utilizados deben ser variados, tratando de abarcar todas las posibles entradas de datos.

**Resultados de los casos de prueba aplicados al ADT**

# Prueba	1	2	3	4
Input ("Escribe el número de elementos de la lista:")	4	0	k 6	*-+ 2
Output esperado	Ingresar dato:	Lista actual: 0	Error: No es un número válido. Intentar de nuevo.  Ingresar dato:	Error: No es un número válido. Intentar de nuevo.  Ingresar dato:
Output código	Ingresar dato:	Lista actual:	Error: No es un número válido. Intentar de nuevo.  Ingresar dato:	Error: No es un número válido. Intentar de nuevo.  Ingresar dato:
Input ("Ingresar dato:")	7 6 8 9	-	j  5 8 9 7 6 4	//*.  1 1
Output esperado	Lista actual: Dato: 7 Dato: 6 Dato: 8	-	Error: No es un número válido. Intentar de nuevo.	Error: No es un número válido. Intentar de nuevo.

	Dato: 9		Lista actual: Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Lista actual: Dato: 1 Dato: 1
Output código	Lista actual: Dato: 7 Dato: 6 Dato: 8 Dato: 9	-	Error: No es un número válido. Intentar de nuevo.  Lista actual: Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Error: No es un número válido. Intentar de nuevo.  Lista actual: Dato: 1 Dato: 1
Input ("Ingresar dato a insertar en inicio:")	7	0	a  3	%%%  1
Output esperado	Dato: 7 Dato: 7 Dato: 6 Dato: 8 Dato: 9	Dato: 0	Error: No es un número válido. Intentar de nuevo.  Dato: 3 Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Error: No es un número válido. Intentar de nuevo.  Dato: 1 Dato: 1 Dato: 1
Output código	Dato: 7 Dato: 7 Dato: 6 Dato: 8 Dato: 9	Dato: 0	Error: No es un número válido. Intentar de nuevo.  Dato: 3 Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Error: No es un número válido. Intentar de nuevo.  Dato: 1 Dato: 1 Dato: 1
Input ("Ingresar elemento a buscar:")	5  6	4  0	g  yy	—  1

			7	
Output esperado	Elemento no encontrado. Intenta de nuevo.  Elemento encontrado en la posición: 2	Elemento no encontrado. Intenta de nuevo.  Elemento encontrado en la posición: 0	Error: No es un número válido. Intentar de nuevo.  Error: No es un número válido. Intentar de nuevo.  Elemento encontrado en la posición: 4	Elemento no encontrado. Intenta de nuevo.  Elemento encontrado en la posición: 0
Output código	Elemento no encontrado. Intenta de nuevo.  Elemento encontrado en la posición: 2	Elemento no encontrado. Intenta de nuevo.  Elemento encontrado en la posición: 0	Error: No es un número válido. Intentar de nuevo.  Error: No es un número válido. Intentar de nuevo.  Elemento encontrado en la posición: 4	Elemento no encontrado. Intenta de nuevo.  Elemento encontrado en la posición: 0
Input ("Ingresa el elemento a actualizar:" & "Ingresar el nuevo elemento:")	7  5	0  0	eli  1  7	1  ?  1
Output esperado	El elemento 7 fue actualizado a 5. Dato: 5 Dato: 7 Dato: 6 Dato: 8 Dato: 9	El elemento 0 fue actualizado a 0. Dato: 0	Error: No es un número válido. Intentar de nuevo.  El elemento 1 no fue encontrado en la lista. Dato: 3 Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Error: No es un número válido. Intentar de nuevo.  El elemento 1 fue actualizado a 1. Dato: 1 Dato: 1 Dato: 1
Output código	El elemento 7 fue actualizado a 5. Dato: 5	El elemento 0 fue actualizado a 0. Dato: 0	Error: No es un número válido. Intentar de	Error: No es un número válido. Intentar de

	Dato: 7 Dato: 6 Dato: 8 Dato: 9		nuevo.  El elemento 1 no fue encontrado en la lista. Dato: 3 Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	nuevo.  El elemento 1 fue actualizado a 1. Dato: 1 Dato: 1 Dato: 1
Input ("Ingresa el elemento a eliminar:")	5	0	programacion  hola  1	1
Output esperado	Elemento 5 eliminado. Dato: 7 Dato: 6 Dato: 8 Dato: 9	Elemento 0 eliminado.	Error: No es un número válido. Intentar de nuevo.  Error: No es un número válido. Intentar de nuevo.  Error: Elemento 1 no encontrado en la lista. Dato: 3 Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Elemento 1 eliminado. Dato: 1 Dato: 1
Output código	Elemento 5 eliminado. Dato: 7 Dato: 6 Dato: 8 Dato: 9	Elemento 0 eliminado.	Error: No es un número válido. Intentar de nuevo.  Error: No es un número válido. Intentar de nuevo.  Error: Elemento 1 no encontrado en la lista. Dato: 3 Dato: 5 Dato: 8	Elemento 1 eliminado. Dato: 1 Dato: 1

			Dato: 9 Dato: 7 Dato: 6 Dato: 4	
Input ("Ingresa la posición en donde insertar:" & "Ingresar dato a insertar:")	1  4	0  0	lol  16  3	##""  1 1
Output esperado	Dato: 7 Dato: 4 Dato: 6 Dato: 8 Dato: 9	-	Error: No es un número válido. Intentar de nuevo.  La posición 16 está fuera de los límites de la lista. Dato: 3 Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Error: No es un número válido. Intentar de nuevo.  Dato: 1 Dato: 1 Dato: 1
Output código	Dato: 7 Dato: 4 Dato: 6 Dato: 8 Dato: 9	Dato: 0	Error: No es un número válido. Intentar de nuevo.  La posición 16 está fuera de los límites de la lista. Dato: 3 Dato: 5 Dato: 8 Dato: 9 Dato: 7 Dato: 6 Dato: 4	Error: No es un número válido. Intentar de nuevo.  Dato: 1 Dato: 1 Dato: 1

Se eligieron estos casos de prueba debido al rango de respuestas que proporcionan. Los datos ingresados son de tipo numérico, de tipo string, de tipo char y caracteres especiales. De igual manera se proporcionaron datos que no eran válidos en el rango de elementos de las listas para verificar el funcionamiento de los try-catch implementados en el código.

### Sección de la calidad de software del ADT

Para evaluar la calidad de software del ADT, se debe analizar la indentación, la facilidad de lectura y entendimiento, junto con la capacidad de manejar errores dentro del código:

Se entiende como indentación a aquella práctica de correcta escritura de la estructura del código. En este caso, se está utilizando una correcta y coherente estructura del código,

donde se logra visualizar la jerarquía de operaciones y bloques de control gracias a la indentación de líneas.

Por su parte, la documentación se entiende como aquella información proporcionada dentro del código. Esta información ayuda a garantizar que futuros colaboradores o desarrolladores que quieran leer y utilizar el código puedan comprender su funcionamiento y el propósito de cada función. Esto se lleva a cabo mediante el uso de comentarios y breves descripciones de comportamiento. El código presentado ofrece comentarios pertinentes en todas las funciones que son relevantes para el entendimiento del código y su funcionamiento.

La mantenibilidad se refiere a la posibilidad de modificar el código con facilidad, ya sea para adaptarlo a otras necesidades o para corregir errores. De forma que un software con buena mantenibilidad es uno que es fácil de actualizar sin generar errores o cambiar el funcionamiento completo del sistema. Debido a que en el código presentado se están utilizando templates y funciones, es posible modificar las funciones de manera individual sin afectar las demás partes del código.

La facilidad de lectura y entendimiento se puede identificar mediante el análisis de la estructura del código. Entre más descriptivo sea, más entendible será. Debido a que en el código todas las funciones tienen nombres identificables, es sencillo comprender las distintas partes del mismo. A su vez, el código está separado por comentarios que vuelven el código más legible.

El ADT de lista enlazada desarrollado cumple con los requisitos esenciales para la manipulación dinámica de datos. Todas las funciones contienen comentarios para profundizar en su funcionamiento, junto con un manejo eficiente de errores. La estructura del código sigue los parámetros del Estándar de Codificación y es visualmente intuitiva. El código es mantenible y comprensible, lo que lo vuelve apto para el uso de quienes deseen utilizarlo en un futuro.