

Por:

Jorge Ivan Coronado Villegas

Introducción

Los ADT (Tipos de Datos Abstractos) son una forma de representar datos y operaciones que se pueden realizar sobre ellos, sin especificar los detalles de cómo se implementan. Los ADT son importantes porque permiten abstraer la complejidad de los datos y encapsular su comportamiento, facilitando el diseño, el desarrollo y el mantenimiento de programas.

Un ejemplo de ADT es la lista enlazada (Linked List), que es una estructura de datos que almacena una secuencia de elementos, donde cada elemento tiene una referencia al siguiente. Las listas enlazadas se utilizan para implementar otras estructuras de datos como pilas, colas, árboles, grafos, etc. También se usan para resolver problemas que requieren inserción y eliminación eficiente de elementos en cualquier posición.

Para implementar CRUD (Create, Read, Update, Delete) con listas enlazadas, se necesita definir una clase **Nodo** que contenga el dato y el puntero al siguiente nodo, y una clase **Lista** que contenga la referencia al primer nodo y métodos para realizar las operaciones CRUD.

En nuestro código se puede observar una implementación de una Lista Enlazada, la aplicación de lo anteriormente dicho y sus operaciones CRUD junto a la complejidad de cada método utilizado de la clase `LinkedList`, aparte de las operaciones CRUD también implementamos otros métodos requeridos por el profesor y además 2 métodos adicionales que creímos oportunos (`cargarLista` y `guardarLista`) los cuales como estamos trabajando con métodos CRUD que son operaciones utilizadas en una base de datos, creímos oportuno implementar la simulación de una base de datos a base de 2 txt, uno que sería el `input(listaEnlazada)` y otro que sería el `output(listaEnlazadaOutput)`.

Casos de Prueba

Input (txt)	Input (teclado)	Output (consola)	Output (txt)
Números del 10 al 590 De 10 en 10	Opción 5 (mostrar lista enlazada) Opción 0 (salir)	“Lista: Números del 10 al 590 De 10 en 10 mostrados en consola.” “Lista Final: Números del 10 al 590 De 10 en 10 mostrados en consola.”	Números del 10 al 590 De 10 en 10
Números del 10 al 590 De 10 en 10	Opción 5 (mostrar lista enlazada) Opción 8 (Borrar lista entera) Opción 0 (salir)	“Lista: Números del 10 al 590 De 10 en 10 mostrados en consola.” “La lista se borro” “La aplicación finalizo con la lista vacia”	Vacio
Borrar todo el texto Vacio	Opción 5 (mostrar lista enlazada) Opción 0 (salir)	“La lista esta vacia” “La aplicacion finalizo con la lista vacia”	Vacio

<p>Vacio</p>	<p>Opción 5 (mostrar lista enlazada)</p> <p>Opción 1 (Insertar) Dato ingresado: "90" Opción 1(Al inicio)</p> <p>Opción 1(Insertar) Dato ingresado: "20" Opción 2(Al final)</p> <p>Opción 1(Insertar) Dato Ingresado: "1" Opción 3(Después de un nodo) Nodo ingresado: "90"</p> <p>Opción 5 (mostrar lista enlazada)</p> <p>Opción 0 (salir)</p>	<p>"La lista esta vacia"</p> <p>"Lista: 90 1 20"</p> <p>"Lista Final: 90 1 20"</p>	<p>90 1 20</p>
<p>*Copiar el texto del output txt anterior y pegar*</p> <p>90 1 20</p>	<p>Opción 5 (mostrar lista enlazada)</p> <p>Opción 2 (Updatear) Nodo ingresado: "1" Dato Actualizado: "2"</p> <p>Opción 5 (mostrar lista enlazada)</p> <p>Opción 3 (Eliminar) Opción 3(Dato exacto) Nodo ingresado: "2"</p> <p>Opción 5 (mostrar lista enlazada)</p> <p>Opción 4 (Buscar) Nodo ingresado: "2"</p> <p>Opción 6 (Primer elemento)</p> <p>Opción 7 (Lista vacia?) Opción 0 (Salir)</p>	<p>"Lista: 90 1 20"</p> <p>"Lista: 90 2 20"</p> <p>"Lista: 90 20"</p> <p>"El nodo no existe"</p> <p>"El primer elemento es: 90"</p> <p>"La lista no esta vacia" "Lista Final: 90 20"</p>	<p>90 20</p>

Las razones por las cuales se escogieron estas pruebas, es porque son pruebas que testean todas las funcionalidades de la aplicación, entonces escogimos los pasos e inputs a seguir para comprobar y resaltar que todas nuestras funcionalidades funcionan y trabajan como lo esperamos, en estos 5 casos de prueba que se pueden ver anteriormente trabajan las 8 funcionalidades implementadas a las que el usuario tiene acceso a través de comandos, en forma de opciones pero también hay más funcionalidades o subfuncionalidades que el usuario no puede ver en forma de opciones como la funcionalidad de guardado y lectura de los archivos txt, pero si se puede observar su funcionamiento a través de las otras funcionalidades implementadas.

Habiendo efectuado las pruebas siguiendo los pasos que hicimos, en nuestro caso comprobamos que funciona tal como lo esperado, tanto los inputs como los outputs.