



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA

UTP
DIRECCIÓN GENERAL DE UNIVERSIDADES
TECNOLÓGICAS Y POLITÉCNICAS

UN
Universidad Tecnológica
del Norte de Guanajuato
Organismo Público Descentralizado del Gobierno del Estado
"Educación y progreso para la vida"



Licenciatura en Ingeniería en Tecnologías de la Información e Innovación Digital

Desarrollo de Software Multiplataforma

Estructura de Datos

Unidad II

U2EJERGUA01 La Lista Encantada – Versión Enlazada

Ejercicio Práctico: Manipulación de una Lista Enlazada Simple en Visualgo

Grupo: GTID141

ALUMNA:

Serna Rodríguez Sara Lizbeth 1224100715



Objetivo

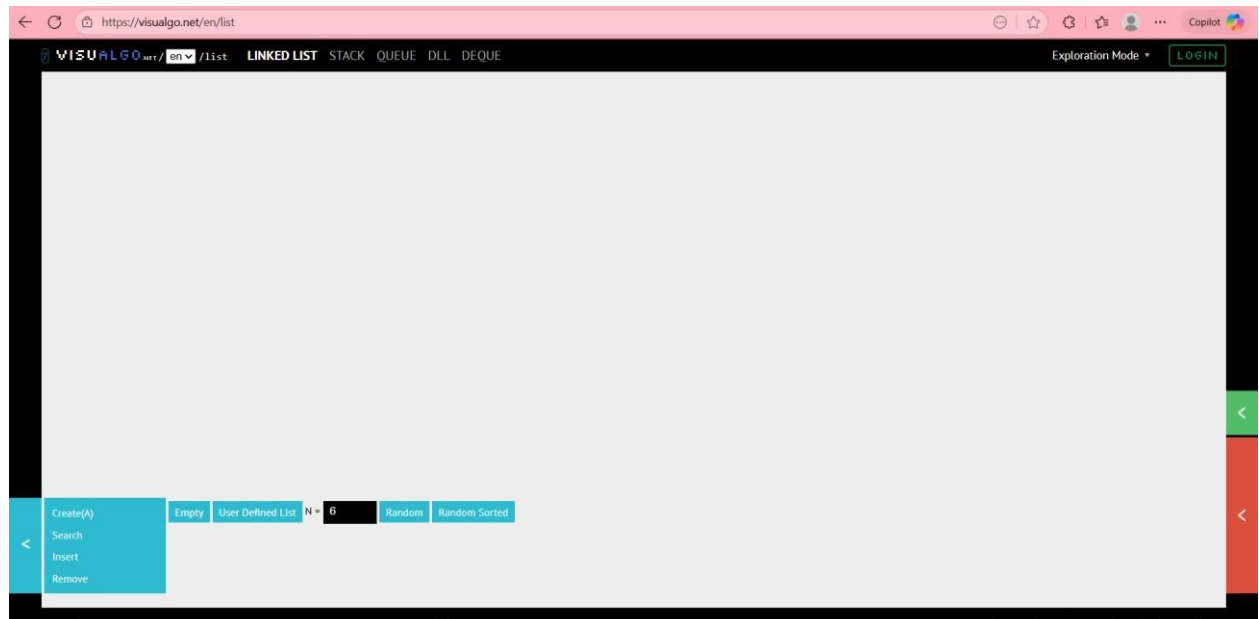
Permite entender visualmente cómo se mueven los punteros o referencias y cómo cambia la estructura con cada operación.

En la plataforma VisuAlgo

visualizamos cómo se comporta una lista enlazada al realizar operaciones como:

- Inserción al inicio, medio o final.
- Eliminación de nodos.
- Búsqueda de elementos.
- Recorrido de la lista.

Crear una lista enlazada vacía.





Insertar elementos en la lista

← ↻ 🔒 <https://visualgo.net/en/list>

VISUALGO [en](#) /list **LINKED LIST** STACK QUEUE DLL DEQUE

```
graph LR; head[head/0] --> n0((22)); n0 --> n1((2)); n1 --> n2((77)); n2 --> n3((6)); n3 --> n4((43)); n4 --> n5((76)); n5 --> n6((89)); n6 --> tail[tail/6];
```

tmp
We d
if
ind
whi
i
i
ret

< Create(A) Search Insert Remove

i = 0 (Head), specify v =
v = 50 Go

i = N (After Tail), specify v =

specify both i in [1..N-1] and v =

Eliminar elementos de la lista



VISUALGO [en](#) /list LINKED LIST STACK QUEUE DLL DEQUE Exploration Mode [LOGIN](#)

head/0 p tmp tail/6

Create(A)
Search
Insert
Remove

Remove i = 0 (Head) Remove i = N-1 (Tail) specify i in [1..N-2]

Remove i = N-1 (Tail)

```
Check if tmp.next is null.  
  
if empty, do nothing  
Vertex pre = head  
tmp = head.next  
while (tmp.next != null)  
    pre = tmp.next; tmp = tmp.next;  
pre.next = null  
delete tmp, tail = pre
```

Buscar elementos en la lista

<https://visualgo.net/en/list> **VISUALGO** [en](#) /list LINKED LIST STACK QUEUE DLL DEQUE Exploration Mode [LOGIN](#)

head/0 tmp/3 tail/6

Search 12

```
So tmp advances to the next vertex.  
tmp is not null, continue searching.  
  
if empty, return NOT_FOUND  
index = 0, tmp = head  
while (tmp.item != v)  
    index++, tmp = tmp.next  
if tmp == null  
    return NOT_FOUND  
return index
```



Preguntas

1. ¿Qué sucede con los punteros cuando se inserta o elimina un nodo?

Al insertar un nodo: El puntero del nodo anterior se actualiza para que apunte al nuevo nodo,
al eliminar un nodo: El puntero del nodo anterior se redirige para que apunte al nodo siguiente del que se elimina.

2. ¿Cómo afecta la posición de un nodo (inicio, medio, final) al tiempo de búsqueda?

La posición de un nodo en una lista enlazada simple influye en el tiempo de búsqueda: los nodos al inicio se localizan rápidamente, mientras que los del medio o final requieren más tiempo, ya que el recorrido es secuencial desde el primer nodo.

3. ¿Qué ventajas tiene recorrer una lista enlazada frente a otras estructuras como arreglos?

las listas enlazadas son ideales cuando se necesita modificar la estructura con frecuencia, y su recorrido permite una comprensión más profunda de cómo se gestionan los datos en memoria.

4. ¿Cómo podrías implementar la comprobación de una lista vacía en un lenguaje de programación como Java?

Para comprobar si una lista enlazada está vacía en Java, se observa visualmente una lista vacía en la plataforma VisuAlgo.net, sin nodos y con punteros nulos. Esta representación se traduce en código Java verificando si el nodo cabeza es nula.