

# **DSViz: Listas enlazadas**

Versión 1.0: Visualización e interacción con listas enlazadas

**Gustavo Gutiérrez-Sabogal**  
**Jovanny Bedoya-Guapacha**  
**Nancy Janet Castillo-Rodríguez**



# Índice

<b>1</b>	<b>Introducción</b>	<b>7</b>
1.1	Instalación	7
<b>2</b>	<b>Funcionalidad</b>	<b>9</b>
2.1	Interfaz de usuario	9
2.2	Visualización de la lista	10
<b>3</b>	<b>Operaciones</b>	<b>13</b>
3.1	Creación	13
	Lista vacía	13
	Lista con números aleatorios	13
	Lista personalizada	14
3.2	Operaciones sobre la lista	14
3.2.1	Inserción	15
3.2.2	Borrado	18
3.2.3	Acceso	19
3.2.4	Búsqueda	19
3.2.5	Algoritmos	20
3.3	Herramientas de programador	20
3.3.1	Apuntadores al primer y último nodo	21
3.3.2	Apuntadores adicionales	21
3.4	Otras operaciones	22
<b>4</b>	<b>Trabajo futuro</b>	<b>23</b>







# 1. Introducción

Las **estructuras de datos** son uno de los pilares del desarrollo de software. Estas definen como la información (los datos) son almacenados de manera eficiente para ser utilizados por los **algoritmos**. Es por esta razón que su enseñanza y aprendizaje constituyen un reto dentro de la carrera de ingeniería de sistemas.

La idea de crear un simulador tiene diferentes fundamentos:

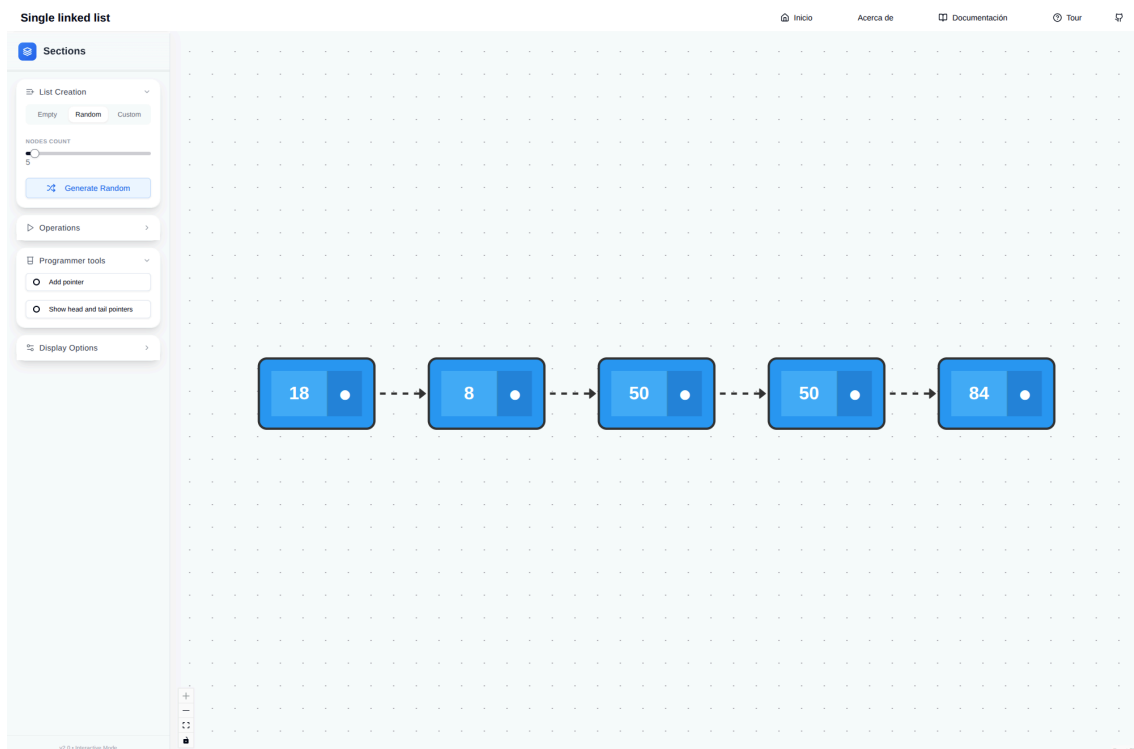
- Proveer a los profesores del curso con una herramienta que permita mostrar ideas y conceptos fundamentales a los estudiantes de una manera clara, precisa y eficaz.
- Permitir a los estudiantes tener un recurso de aprendizaje que puedan utilizar durante la clase y también de manera asincrónica para reforzar, entender y verificar los conceptos impartidos durante el curso.
- Recopilar lo que por muchas iteraciones del curso de estructuras de datos se ha constituido como material de estudio. Esto no solo comprende código fuente sino también valoración sobre los aspectos particulares de cada estructura de datos en donde se ha evidenciado que nuestros estudiantes presentan mayores inconvenientes.

## 1.1 Instalación

El software aquí descrito se puede acceder a través del siguiente enlace:

<https://gustavogutierrezutp.github.io/AlgorithmVisualizer/sll>

Después de hacer click en el enlace, se abrirá el navegador en una página como la que se presenta a continuación.



Como se trata de una herramienta pedagógica que tiene como objetivo ser fácilmente accesible los autores optaron por la manera más sencilla y con cero requisitos de instalación. Todo corre completamente en el navegador y no requiere de ninguna otra dependencia. En el Capítulo 2. se describe en detalle el funcionamiento de la aplicación.





## 2. Funcionalidad

Antes de comenzar a describir cada una de las funcionalidades de la aplicación es importante describir la interfaz de usuario y las partes que la componen. Cuando el usuario (estudiante o profesor) ejecuta la aplicación por primera vez se encuentra con una interfaz como la presentada en la Figura 2.1.

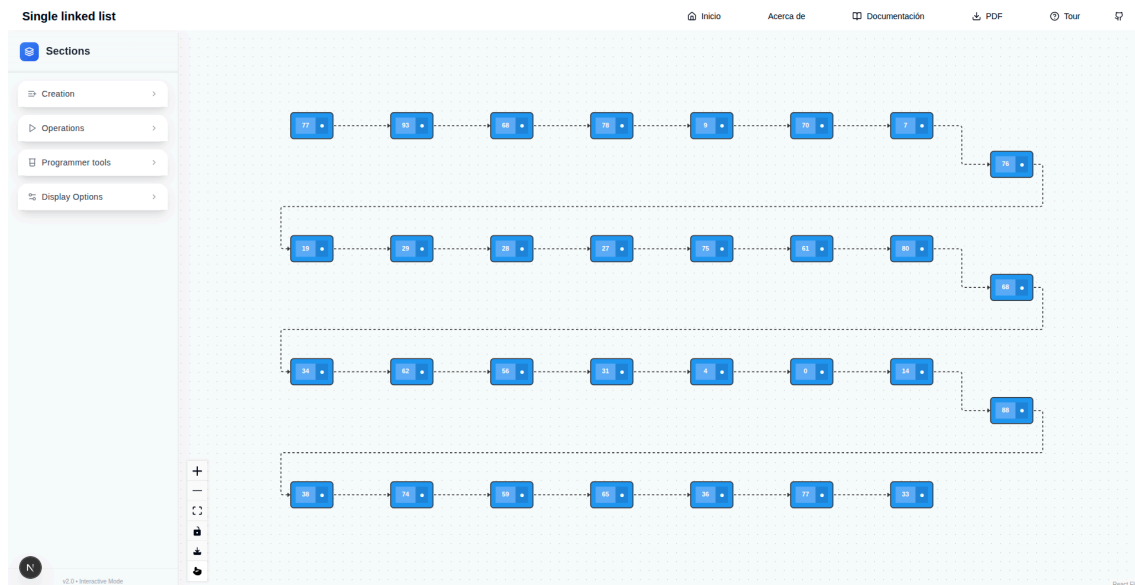


Figura 2.1: Ventana principal con lista de ejemplo desplegada.

### 2.1 Interfaz de usuario

La Figura 2.2 presenta cada una de las partes en las que está dividida la interfaz de usuario:

1. Barra de navegación. Aquí se encuentran enlaces a la documentación y al repositorio con el código fuente. También un enlace a un tour que lleva al usuario por cada uno de los componentes y funcionalidades más importantes de la aplicación.
2. Lienzo. Esta es la parte donde toda la visualización tiene lugar. Es el lugar donde se visualizará la lista y se presentarán las animaciones de las operaciones según sean seleccionadas por el usuario.
3. Menú de operaciones. En este lugar se encuentran de manera agrupada por funcionalidad las acciones que el usuario puede realizar en la aplicación.

El flujo de trabajo que se espera por parte del usuario está enfocado a utilizar las partes 2 y 3 de la interfaz de usuario. La idea es que este último selecciona la operación en la parte 3 y visualiza su ejecución en la parte 2.



Figura 2.2: Partes principales de la interfaz de usuario.

Las operaciones soportadas por la aplicación están divididas por funcionalidad y presentadas de esta manera en la parte 3:

- Creación. Comprenden las diferentes formas en que se puede crear una lista enlazada. Entre estas formas se incluye: vacía, de una secuencia de números o de manera aleatoria.
- Operaciones sobre la estructura de datos. Son las operaciones que le permiten al programador interactuar con la estructura de datos. En esta sección se incluyen operaciones de inserción, acceso y eliminación entre otras.
- Herramientas de programación. Permiten la visualización de componentes de programación como punteros. Estos elementos no son conceptuales pero si son relevantes a la hora de la implementación de la estructura de datos.
- Opciones de visualización. Permiten al usuario alterar algunas características de la visualización de la estructura de datos. Estas opciones son útiles para la presentación y visualización de la estructura de datos en entornos de estudio o de docencia.

A continuación describiremos cada una de las partes. La parte superior contiene una barra con el título y algunos enlaces de interés. Tal vez el más importante es el botón con el texto **Tour** en la parte superior derecha. Pulsar este botón activará una especie de tour por cada uno de los elementos de la interfaz gráfica y le proporcionará al usuario una idea de la funcionalidad de cada parte.

Las otras dos partes son las más importantes para la interacción con el programador. En la sección vertical izquierda se puede apreciar una barra de menús. En el caso de la figura todos ellos se encuentran colapsados y entre otros, presentan los títulos: **List Creation**, **Operations**, **Programmer Tools** y **Display Options**. Finalmente en la sección vertical derecha se encuentra el lienzo que es el lugar donde aparecen las representaciones gráficas. En este caso se aprecia una lista de números. En las secciones siguientes se muestra detalladamente cada una de las funcionalidades.

## 2.2 Visualización de la lista

Después de creada una lista, está será visualizada en el lienzo que hay en la parte derecha. Por ejemplo, en la Figura 2.3 se muestra la interfaz con una lista creada de 7 elementos. En esta figura se encuentran las anotaciones de los diferentes nodos de la lista y los componentes de uno de ellos.

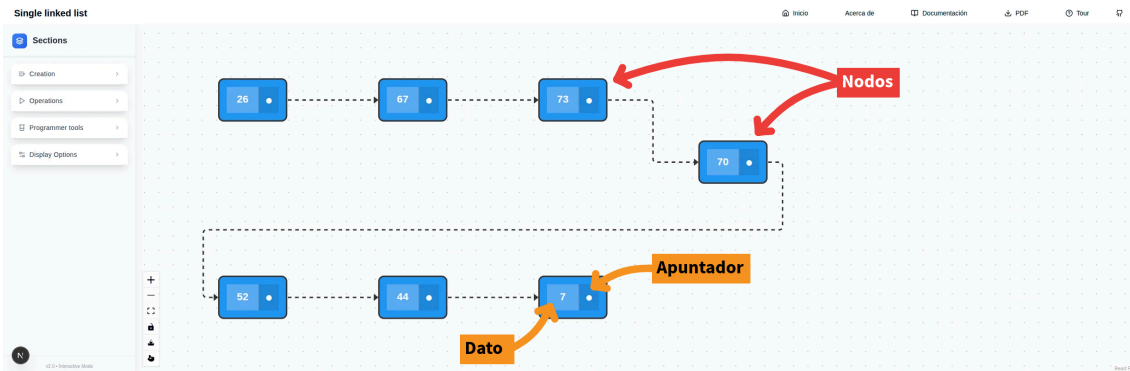



Figura 2.3: Imagen anotada de la visualización de una lista enlazada de 5 elementos en el lienzo.

Existen diferentes formas de interactuar con la representación visual de la lista enlazada. Los nodos de la lista pueden ser reposicionados y su encadenamiento puede ser alterado. En la parte inferior izquierda del lienzo se encuentra una pequeña barra vertical con botones que tienen acción sobre la visualización del lienzo.

#### 💡 Tip 2.1: Consejos



- El tercer botón de arriba a abajo permite siempre enfocar todos los elementos dentro del lienzo. Esto es de gran importancia porque facilita mucho la interacción con la visualización.
- El cuarto botón con el símbolo  protege el lienzo de cualquier tipo de interacción. Esta opción es muy útil cuando se desean tomar imágenes de captura de pantalla o realizar explicaciones sin que se altere la visualización.

Cada nodo de la lista consta de dos partes como se identifican en la Figura 2.3. La primera contiene el dato y la segunda representa el apuntador al siguiente nodo en la lista. Cuando el *mouse* se posiciona sobre este último el apuntador es resaltado en un color diferente (ver Figura 2.4). Esto es útil cuando las listas no se muestran de manera lineal. A continuación se muestra la misma lista después de ser reorganizada arrastrando los nodos con el mouse sobre el lienzo. El puntero se posicionó sobre el nodo que contiene el valor 44.

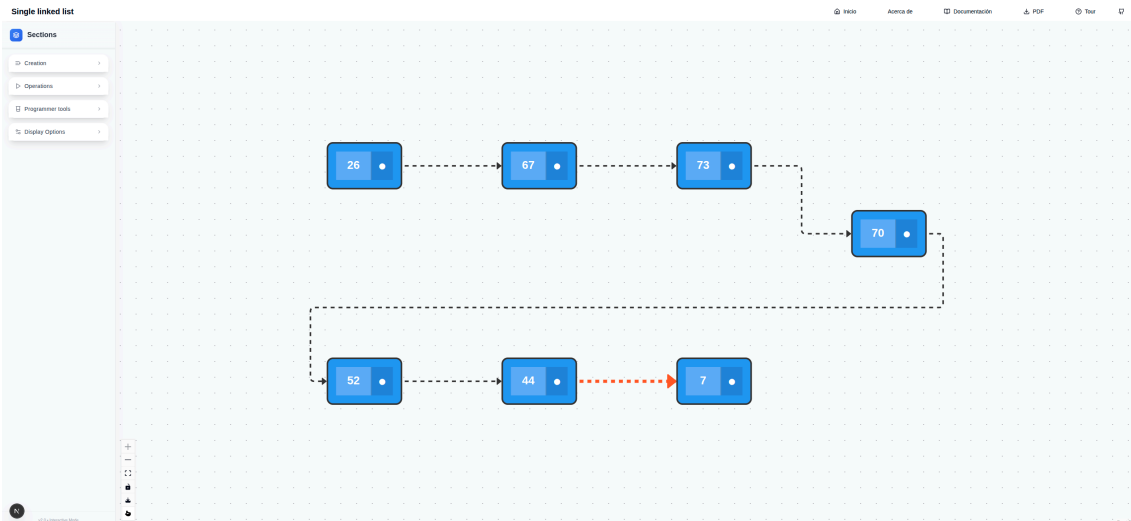


Figura 2.4: Visualización del puntero al siguiente elemento de un nodo. Esta visualización se logra al colocar el puntero del ratón sobre el apuntador del nodo.

## 3. Operaciones

En este capítulo se detallan cada una de las operaciones disponibles a través de la interfaz de usuario.

### 3.1 Creación

En el simulador las listas pueden ser creadas de diferentes maneras. Todas las opciones de creación se encuentran en el primer sub menú con el título **List Creation**.

#### Listas vacías

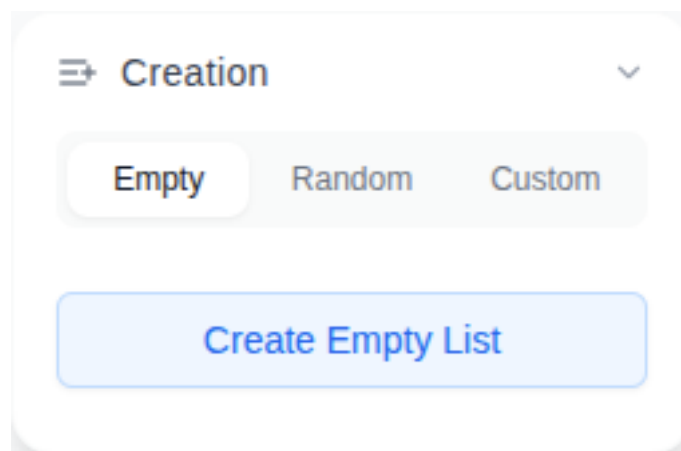


Figura 3.1: Interfaz para la creación de una lista vacía.

Para crear una lista vacía donde el usuario pueda comenzar a añadir elementos haciendo uso de las operaciones se debe hacer click en la pestaña *Empty* y a continuación hacer click en el botón *Create Empty List* (ver Figura 3.1). Al dar click en él aparecerá un mensaje en el lienzo de la derecha informando que la lista actualmente visualizada se encuentra vacía.

#### Listas con números aleatorios

La siguiente opción para crear una lista es con elementos aleatorios. Esto es especialmente útil cuando el usuario desea ver como se comporta una operación sobre una lista que ya contiene elementos y no desea insertarlos uno a uno. En esta opción es cuestión de seleccionar el número de nodos utilizando el elemento deslizable y posteriormente pulsar el botón *Generate Random* (ver Figura 3.2). Al presionar el botón aparecerá la lista en el lienzo de la derecha junto con sus elementos.

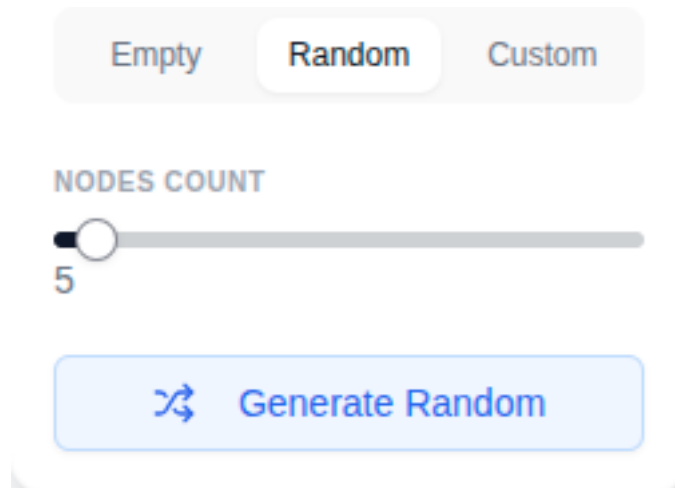


Figura 3.2: Interfaz para la creación de una lista inicializada de forma aleatoria.

### Lista personalizada

Cuando se requiere una lista con elementos específicos la tercera opción es la indicada. Haciendo click en *Custom* aparecerá un cuadro de texto para insertar cada uno de los elementos de la lista (ver Figura 3.3). El formato es simple: los elementos deben estar encerrados por los símbolos `[` y `]`. Adicionalmente los números deben estar separado por comas. Por ejemplo, al introducir `[1,2,3,4,5]` se creará y visualizará la lista que contiene los elementos del 1 al 5.

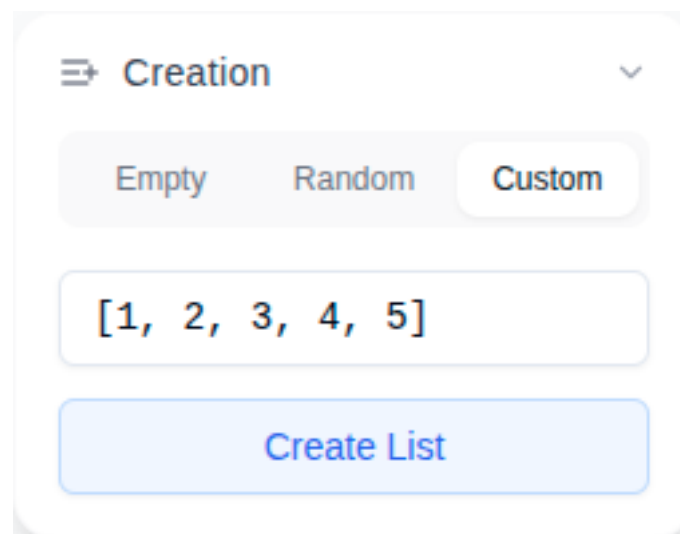


Figura 3.3: Interfaz para la creación de una lista con elementos puntuales.

## 3.2 Operaciones sobre la lista

Después de tener la lista creada o una lista vacía podemos comenzar a realizar operaciones sobre ella. Esto con el fin de observar el comportamiento de cada una de ellas. Las operaciones disponibles se pueden acceder desde el menú **Operations** (ver Figura 3.4).

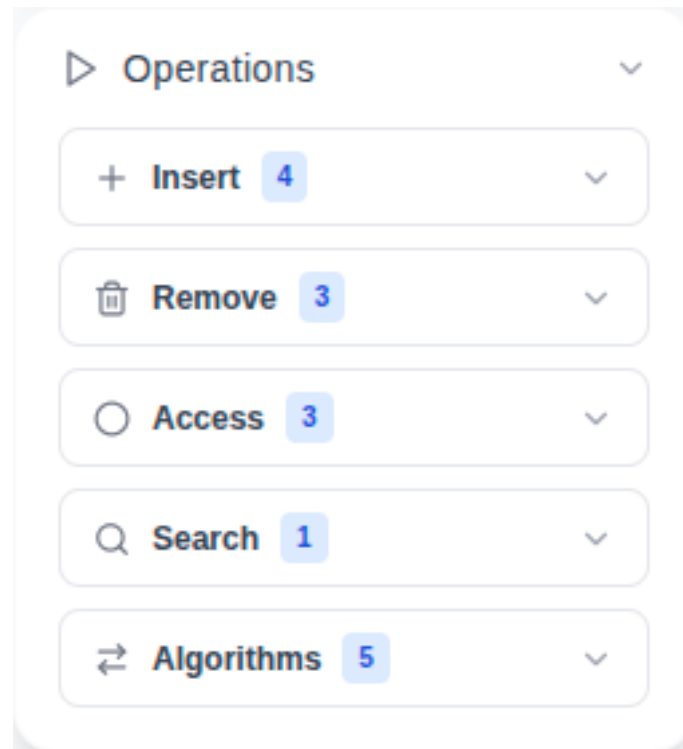


Figura 3.4: Panel de operaciones

Las operaciones se encuentran agrupadas según su lógica. Por ejemplo, el menú **Insert** agrupa las 4 operaciones que adicionan elementos en la lista. El menú **Access** contiene 3 operaciones que acceden a los elementos de la lista y así sucesivamente.

A continuación se describen en detalle cada una de las operaciones y su visualización.

### 3.2.1 Inserción

La aplicación provee visualización para 4 diferentes tipos de inserción a través del menú **insert** mostrado en la figura Figura 3.5. Para todos los casos, el valor a insertar debe ser definido en el primer cuadro de texto. En la figura este presenta el valor 25. También dentro del mismo cuadro de texto hay un botón que opcionalmente genera un valor aleatorio (ver Tip 3.1).

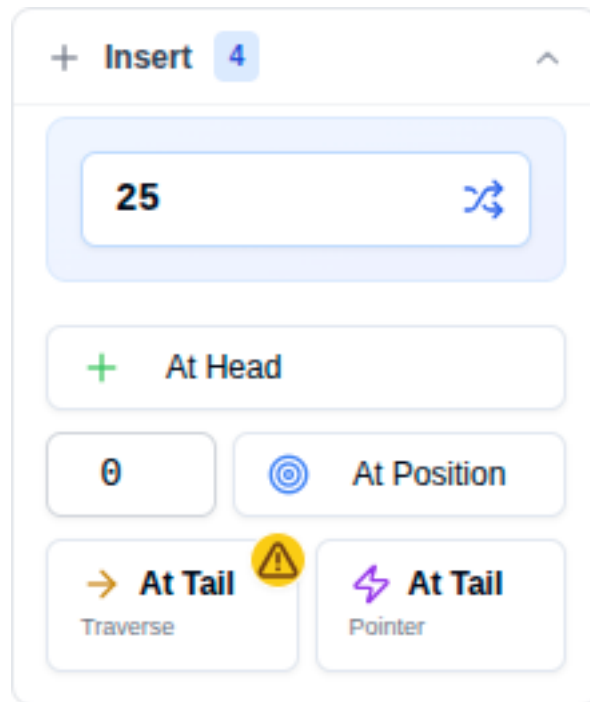


Figura 3.5: Operaciones de inserción.

💡 Tip 3.1: Consejo



Pulsando este botón que se encuentra al lado del cuadro de texto se generará un valor aleatorio. Este valor será insertado por cualquiera de las operaciones.

### At Head

Pulsando el botón **At Head** se inserta el nuevo elemento al inicio de la lista. El nuevo nodo se posicionará al inicio de la lista y tendrá un color diferente (ver Tip 3.2). La Figura 3.6 muestra el lienzo antes de la inserción. Finalmente la Figura 3.7 muestra el lienzo después de la inserción.



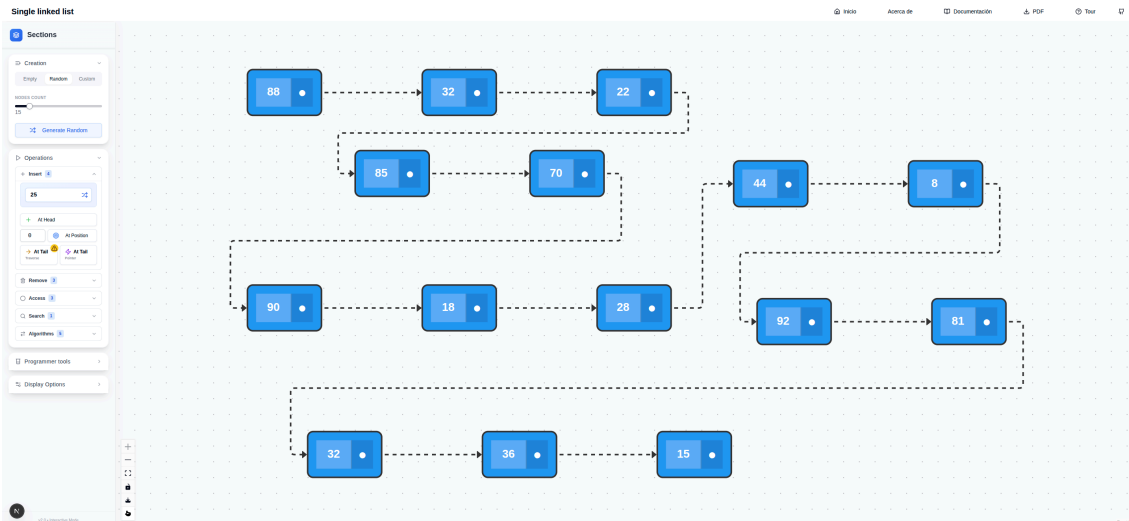


Figura 3.6: Vista anterior a la inserción del valor 25 al inicio de la lista.

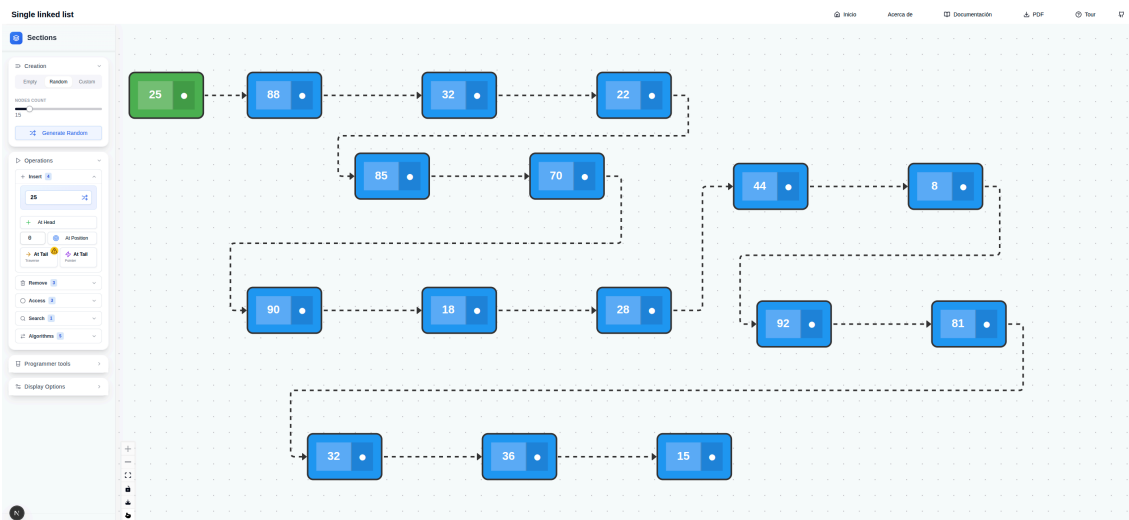


Figura 3.7: Vista posterior a la inserción del valor 25 al inicio de la lista.

### 💡 Tip 3.2: Consejo

Standard Node

Apply

En cualquier momento es posible cambiar el color de los nodos de la lista utilizando la funcionalidad del menú **Display Options** presentado a la izquierda. En este caso se seleccionan en el lienzo los nodos a los que se les desea cambiar el color. Posteriormente se selecciona el color que aparece al lado del botón **Apply** y se pulsa este último.

## At Position

Para insertar un elemento en una posición específica se debe seleccionar la posición deseada en el cuadro de texto al lado del botón **At Position**. Posteriormente se debe pulsar el botón mencionado para insertar el

elemento en la posición seleccionada. En caso de ingresar una posición mayor a la cantidad de elementos de la lista, la entrada de texto se pondrá de color rojo indicando el error (ver Figura 3.8).

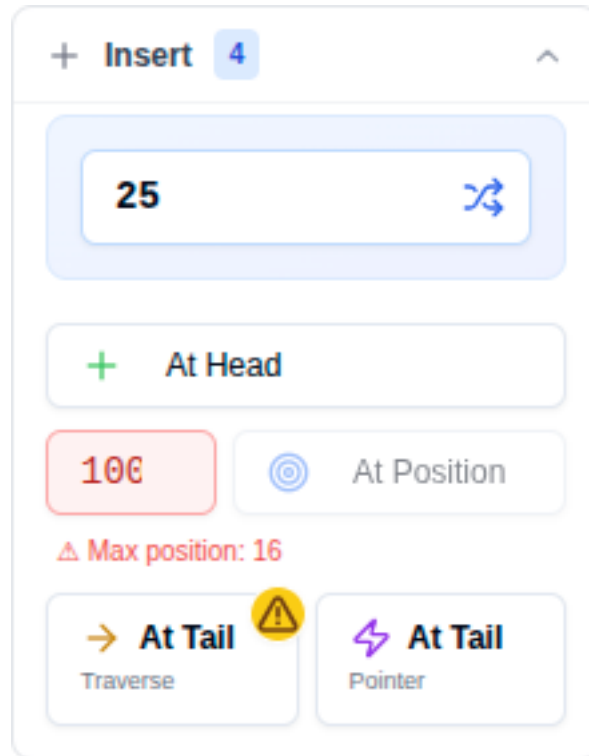


Figura 3.8: Error al elegir una posición no válida

### At Tail

Para insertar un elemento al final de la lista existen dos formas. En ambos casos los botones tienen el mismo nombre: `__At Tail`. Uno de ellos tiene un símbolo de advertencia en la parte superior derecha y el otro no. Ambos insertan el elemento al final de la lista. La diferencia es que uno lo hace recorriendo la lista desde el primer elemento y el otro lo hace utilizando la referencia al último elemento de la lista. Cuando se usa la primera opción se verá la animación en el lienzo de como la lista es recorrida elemento a elemento.

### 3.2.2 Borrado

Para el borrado de elementos existen tres formas diferentes y se muestran en Figura 3.9. El botón con título **Head** permite borrar el primer elemento de la lista. De manera similar el botón con título **Tail** permite borrar el último elemento de la lista. Por último el botón con título **Position** permite borrar un elemento en una posición específica. En este caso primero se debe seleccionar la posición deseada en el cuadro de texto al lado del mismo.

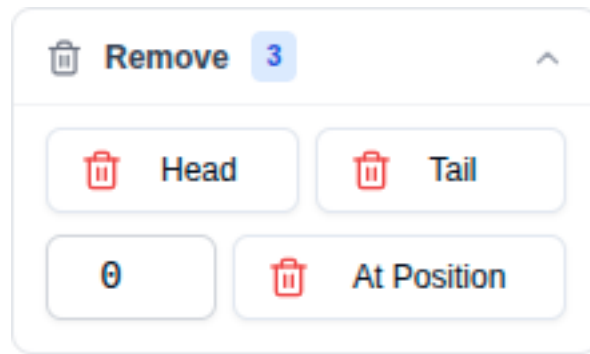


Figura 3.9: Operaciones de borrado.

Cuando se borra utilizando una posición no válida, el cuadro de texto se pondrá de color rojo indicando el error de manera similar al caso de la inserción.

### 3.2.3 Acceso

El siguiente grupo de operaciones es el de acceso. Las operaciones de acceso son tres:

- **Front**: el primer elemento de la lista.
- **Back**: el último elemento de la lista.
- **Nth**: el elemento en una posición específica.

Para cada una de las operaciones el elemento se identificará de manera gráfica en el lienzo.

La Figura 3.10 muestra las operaciones de acceso descritas anteriormente. En el caso de **Nth** es necesario insertar la posición en el cuadro de texto. Cuando se da click en el botón, se visualizará el recorrido hasta llegar a la posición deseada. Si se inserta una posición no válida, el cuadro de texto se pondrá de color rojo indicando el error de manera similar al caso de la inserción.

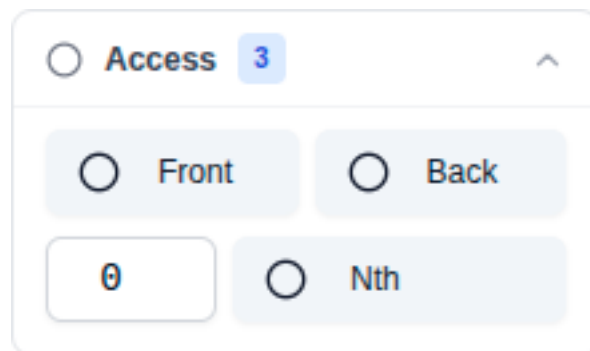


Figura 3.10: Operaciones de acceso a elementos.

### 3.2.4 Búsqueda

La Figura 3.11 contiene la única operación de búsqueda en una lista enlazada. En el cuadro de texto se inserta el valor a buscar. Al pulsar el botón se visualizará de manera animada el recorrido de la lista hasta llegar al valor buscado. Si el valor no se encuentra en la lista, el cuadro de texto se pondrá de color rojo indicando que el elemento no fue encontrado. Si el valor es encontrado entonces el nodo que lo contiene se pondrá de color verde y en el botón aparecerá la posición.

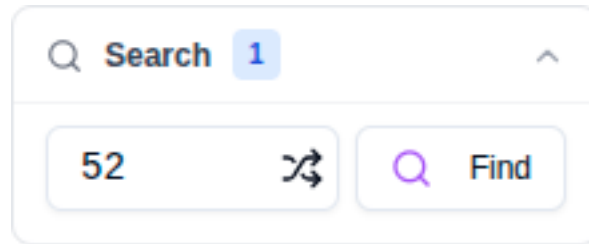


Figura 3.11: Operaciones de búsqueda.

### 3.2.5 Algoritmos

En las secciones anteriores se describieron las operaciones básicas sobre las listas enlazadas. En esta sección se presentan los algoritmos o construcciones más complejas que actúan sobre las listas. Estos se basan en las operaciones básicas y proveen un nivel de abstracción superior.

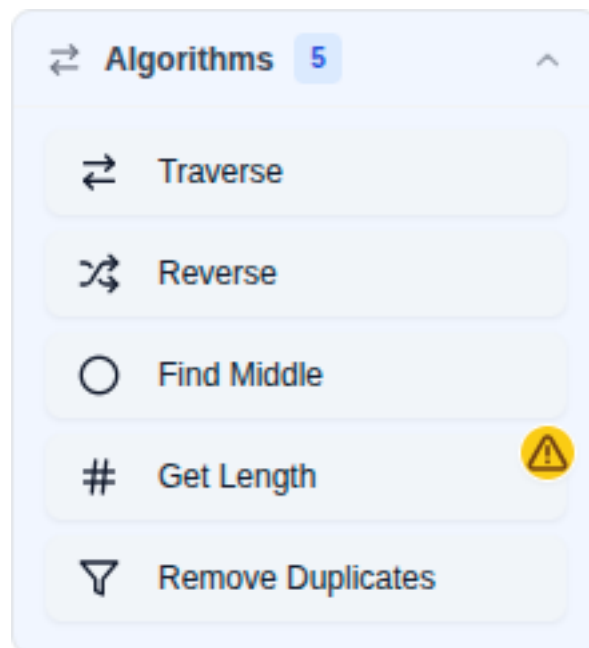


Figura 3.12: Algoritmos sobre listas.

1. **Traverse**: recorre cada uno de los elementos de la lista. Durante su ejecución, el elemento actual será visualizado de un color diferente.
2. **Reverse**: invierte el orden de los elementos de la lista.
3. **Find Middle**: muestra visualmente la ejecución del algoritmo que encuentra el elemento que hay en la mitad de la lista.
4. **Get Length**: recorre la lista para contar el número de elementos. Este algoritmo tiene el símbolo de advertencia porque, aunque es bueno desde un punto de vista pedagógico, no se suele usar ya que el número de elementos en la lista se mantiene como parte de su estado.
5. **Remove Duplicates**: Muestra de manera animada como se encuentran y eliminan los elementos duplicados de una lista.

## 3.3 Herramientas de programador

Una característica importante que tiene este módulo de visualización es la separación de conceptos. Por ejemplo, una lista, conceptualmente, es una secuencia de elementos. A nivel de programación surgen algunos conceptos que son importantes a la hora de implementar esta secuencia. Por ejemplo, se hace

necesario hablar de punteros y direcciones de memoria. Por esta razón el simulador incluye un conjunto de herramientas en el menú **Programmer Tools** que se muestra en la Figura 3.13.

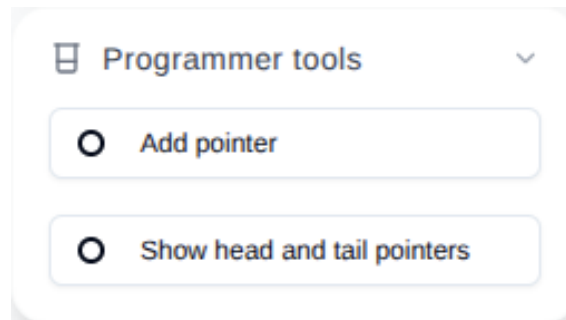


Figura 3.13: Vista de herramientas de programador.

### 3.3.1 Apuntadores al primer y último nodo

El botón **Show head and tail pointers** hará visibles los punteros al primer y último elemento en la lista. Hasta ahora las listas que se han mostrado como la de la Figura 2.4 no tienen esta opción activada. Es posible apreciar la diferencia con la lista que se muestra en la Figura 3.14. Los nodos naranja representan cada uno de los punteros. Si el usuario localiza el puntero del mouse sobre alguno de ellos la conexión se resaltará.

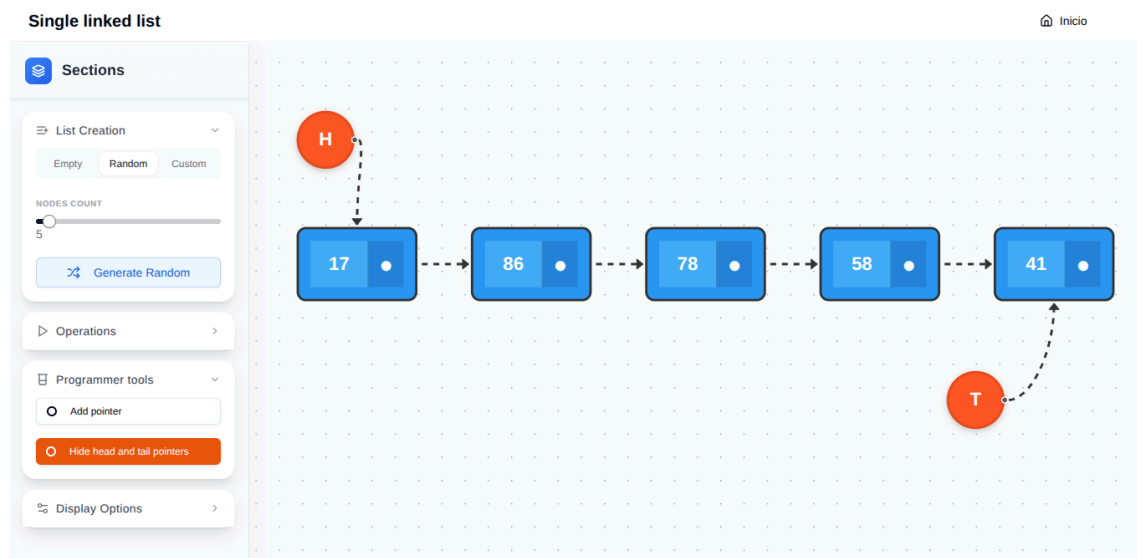


Figura 3.14: Visualización de los punteros al inicio y al final de la lista.

### 3.3.2 Apuntadores adicionales

Durante el diseño de una operación sobre listas es siempre importante considerar el movimiento de los punteros. Incluso operaciones como remover un elemento en una posición de la lista usan punteros adicionales. Para este tipo de operaciones la aplicación cuenta con la funcionalidad de crear nodos de tipo apuntador. Esto se logra haciendo click en el botón **Add pointer** (ver Figura 3.13). Cuando se pulsa este botón se adiciona un nuevo nodo circular al lienzo y este se puede conectar a cualquiera de los nodos de la lista.

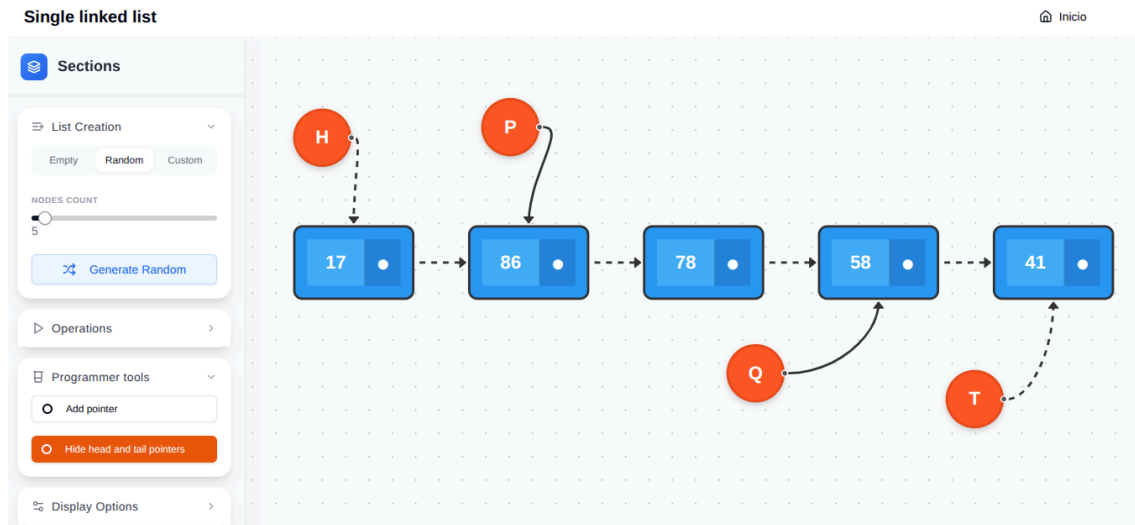




Figura 3.15: Visualización de los punteros adicionales insertados por el usuario.

La Figura 3.15 muestra una lista que incluye los punteros a la cabeza y a la cola. Adicionalmente también hay dos punteros llamados *P* y *Q* que apuntan al segundo y al cuarto elemento.

Los punteros pueden ser reposicionados utilizando el mouse y arrastrando la conexión a otro nodo. Haciendo doble click sobre el nodo es posible cambiar el nombre del puntero. Esto es muy útil en labores de enseñanza.

### 3.4 Otras operaciones

La aplicación cuenta con algunas funcionalidades que no son propias de las estructuras de datos pero si lo son de la enseñanza y el aprendizaje. Por ejemplo, en el Tip 2.1 se observa la barra de herramientas que tiene el lienzo en su parte inferior izquierda.

- El botón  exporta el lienzo en su estado actual a un archivo de imagen con extensión **png**.
- El botón  activa la opción de puntero, la cual actúa como un señalador y ayuda a la comunicación efectiva de las ideas. El puntero del mouse va a estar acompañado de un punto rojo de color atractivo. Para desactivarla se debe dar click de nuevo en el mismo botón.

## 4. Trabajo futuro

A continuación se presentan algunas ideas que pueden ser desarrolladas como trabajo futuro en la forma de versiones posteriores.

1. Lasser pointer
2. Drawing with tablet on the canvas