

Caidas Vazquez Sierra GT10141

17-10-25

EJERCICIO PRÁCTICO: Manipulación de una Pila en Visualgo.net

El método Push sirve para insertar elementos.

Al ser una pila los elementos van ingresándose hasta arriba, y actualiza los punteros, dejando como 0 al nodo que se ingresa al último.

Pop es el método para eliminar elementos.

Elimina el último elemento ingresado. Vuelve temporalmente al elemento que se eliminará, para después cambiar el valor de la cabeza y modificar los punteros, para finalmente eliminar el nodo.

Peek es un método que nos ayuda a identificar la cabeza de la pila, es decir, el último elemento insertado. En Visualgo nos resalta en un color la cabeza de la pila y así poder observarla mejor.

Preguntas de reflexión:

¿Qué estructura se forma visualmente con los nodos?

R= Se va formando una torre que crece hacia arriba.

¿Qué sucede si haces pop en una pila vacía?

R= Si solo tenemos la cabeza, esta será eliminada, si no contiene nada, no elimina nada.

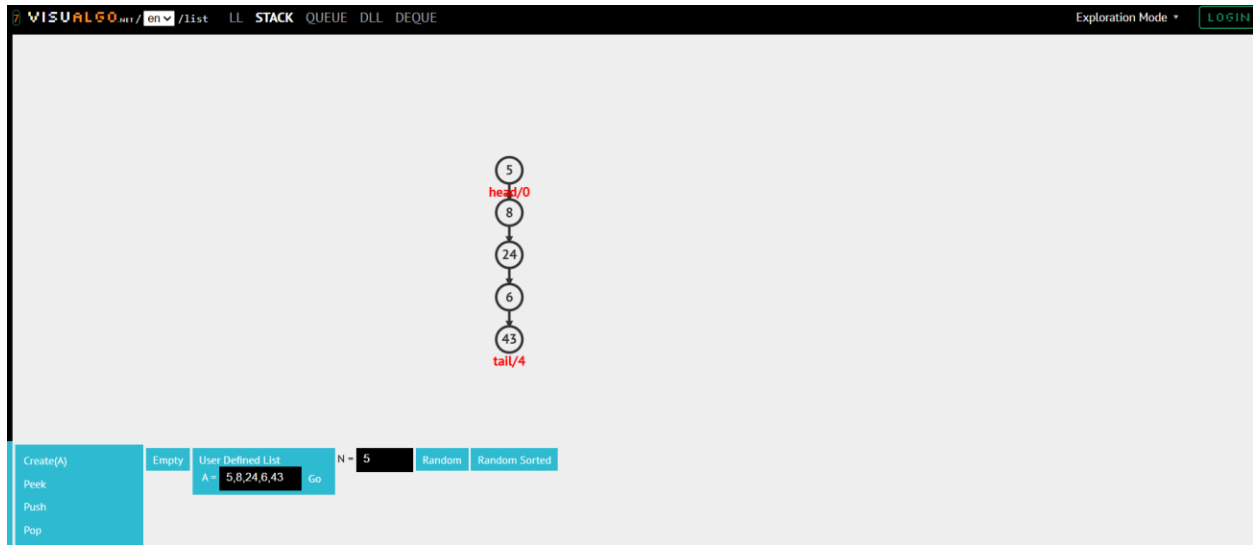
¿Qué diferencia hay entre push y pop?

R= Push nos ayuda a insertar elementos colocándose sobre toda la pila, mientras que pop elimina elementos, pero en orden, eliminando de primero el último elemento ingresado.

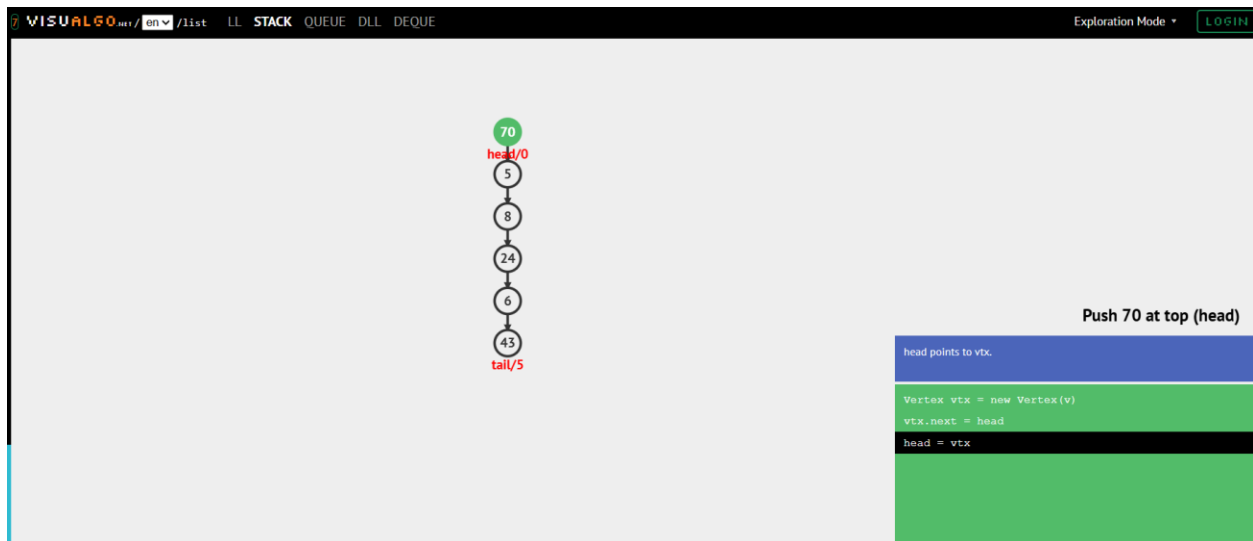
Menciona al menos cinco casos prácticos en donde se utilicen las pilas.

- Funciones de "desahacer" en editores de texto.
- Historial de navegación web.
- Recursividad.
- Calculadoras que almacenan los últimos operaciones realizadas.
- Control remoto que guarda el canal anterior al que se cambió.

Crear Pila



Push con el Nodo(70)



Push con el Nodo(30)

VISUALGO.net /list LL STACK QUEUE DLL DEQUE Exploration Mode LOGIN

```
graph TD; 30((30)) --> 70((70)); 70 --> 5((5)); 5 --> 8((8)); 8 --> 24((24)); 24 --> 6((6)); 6 --> 43((43)); head[head/0] --> 30; tail[tail/6] --> 43;
```

Push 30 at top (head)

```
head points to vtx.  
Vertex vtx = new Vertex(v)  
vtx.next = head  
head = vtx
```

Pop (eliminar ultimo Nodo ingresado)

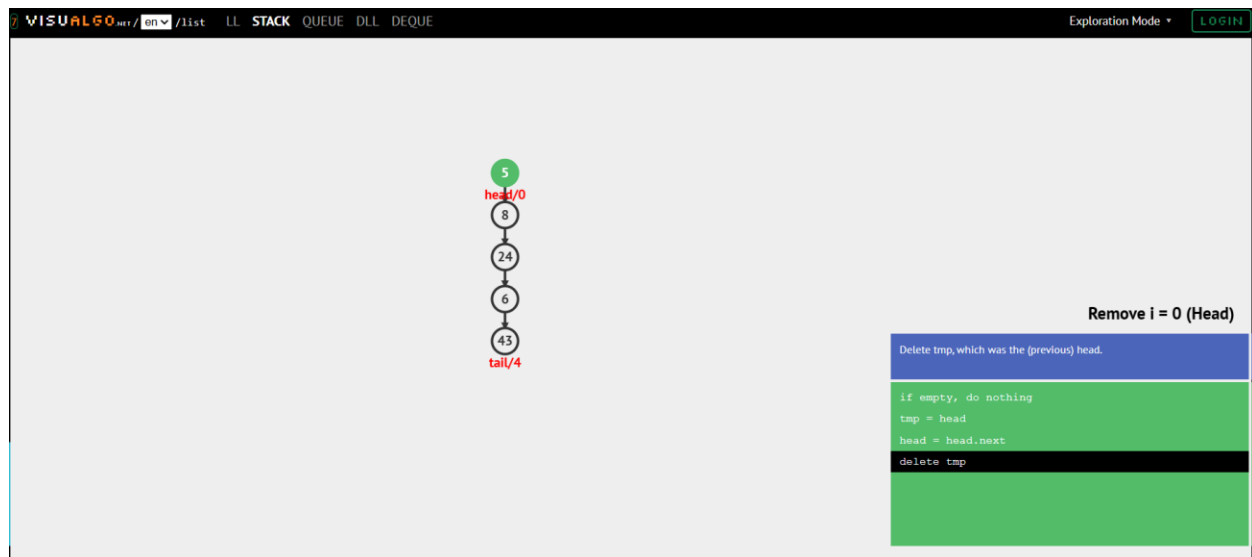
VISUALGO.net /list LL STACK QUEUE DLL DEQUE Exploration Mode LOGIN

```
graph TD; 70((70)) --> 5((5)); 5 --> 8((8)); 8 --> 24((24)); 24 --> 6((6)); 6 --> 43((43)); head[head/0] --> 70; tail[tail/5] --> 43;
```

Remove i = 0 (Head)

```
Delete tmp, which was the (previous) head.  
if empty, do nothing  
tmp = head  
head = head.next  
delete tmp
```

Pop (eliminar último Nodo ingresado)



Peek (resalta el último Nodo, el nodo superior)

