PILAS

Yaneth Mejía Rendón

OBJETIVOS

- Define qué es una pila
- Comprender los casos de uso de una pila.
- Implementar operaciones en una estructura de datos de pila

¿Qué es una pila?

Una estructura de datos **LIFO** !

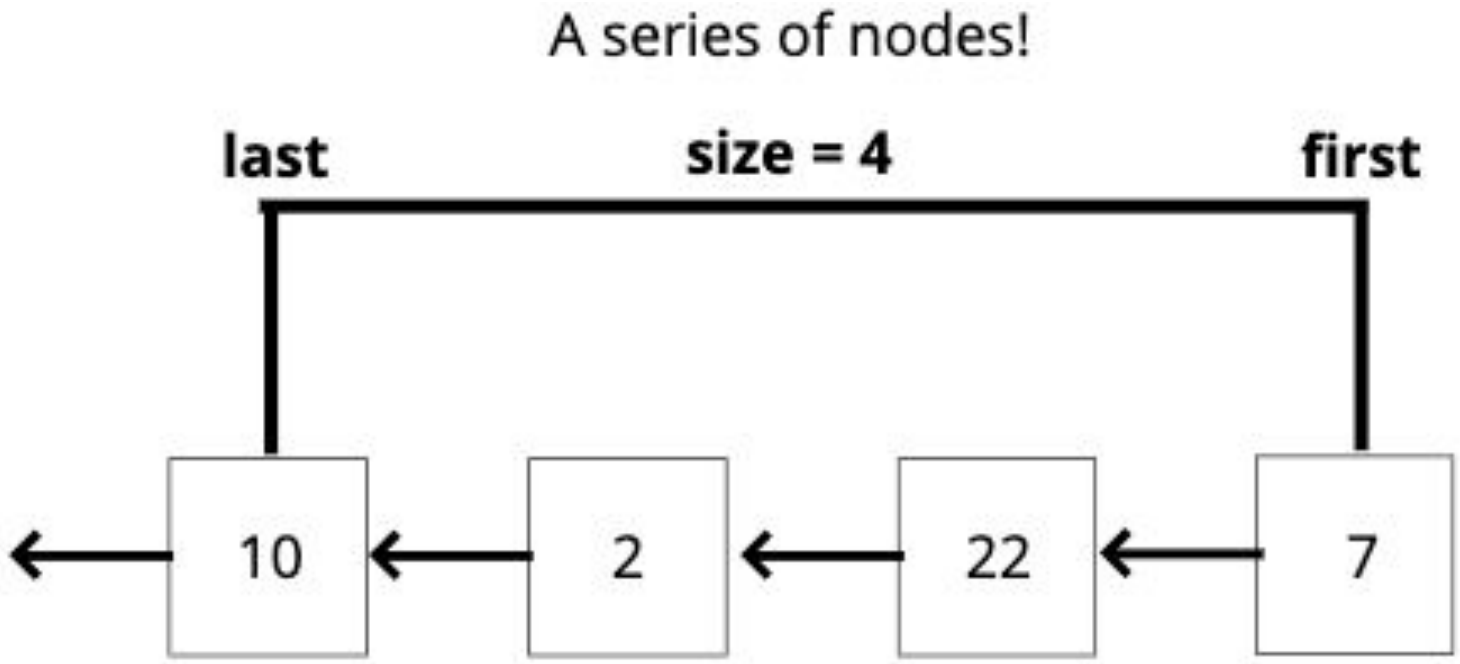
El último elemento agregado a la pila será el primer elemento eliminado de la pila

¿CÓMO SE USA?

Piense en una pila de **platos** , una pila de **marcadores** , o una pila de ... **cualquier cosa** .

A medida que lo acumulas, lo último (o lo que está más arriba) es lo que se elimina primero.

Cómo visualizamos una pila.



HEMOS VISTO ESTO ANTES

La pila de llamadas!

Donde se usan las pilas

- Gestión de invocaciones de funciones.
- Deshacer rehacer
- ¡El enrutamiento se trata como una pila!

HAY MÁS DE UNA MANERA DE IMPLEMENTAR UNA PILA

Implementación de arrays

LISTA DE ENLACE IMPLEMENTACIÓN

Una clase de pila

```
class Stack {
    constructor(){
        this.first = null;
        this.last = null;
        this.size = 0;
    }
}
class Node {
    constructor(value){
        this.value = value;
        this.next = null;
    }
}
```

PUSH

¡**Añade** un valor a la **parte superior** de la pila!

push pseudocodigo

- La función debe aceptar un valor.
- Crea un nuevo nodo con ese valor
- Si no hay nodos en la pila, establezca la primera y la última propiedad como el nodo recién creado
- Si hay al menos un nodo, cree una variable que almacene la primera propiedad actual en la pila
- Restablecer la primera propiedad para ser el nodo recién creado
- Establezca la siguiente propiedad en el nodo como la variable creada anteriormente
- Incrementa el tamaño de la pila en 1.

```
class Node {
    constructor(value){
        this.value = value;
        this.next = null;
    }
}

class Stack {
    constructor(){
        this.first = null;
        this.last = null;
        this.size = 0;
    }
```

```
push(val){
        var newNode = new
Node(val);
        if(!this.first){
            this.first = newNode;
            this.last = newNode;
        } else {
            var temp = this.first;
            this.first = newNode;
            this.first.next = temp;
        }
        return ++this.size;
    }
}
```

POP

¡**Quita** un valor de la **parte superior** de la pila!

PSEUDOCODE POP

- Si no hay nodos en la pila, devuelve null
- Cree una variable temporal para almacenar la primera propiedad en la pila
- Si solo hay 1 nodo, establezca la primera y la última propiedad como nulas
- Si hay más de un nodo, establezca que la primera propiedad sea la siguiente en la primera actual
- Disminuir el tamaño en 1
- Devuelve el valor del nodo eliminado.

```
pop(){
        if(!this.first) return null;
        var temp = this.first;
        if(this.first === this.last){
            this.last = null;
        }
        this.first = this.first.next;
        this.size--;
        return temp.value;
    }
```

BIG O de STACKS

Inserción - **O (1)**

Eliminación - **O (1)**

Buscando - **O (N)**

Acceso - **O (N)**

RESUMEN

- Las pilas son una estructura de datos **LIFO** donde el último valor de entrada es siempre el primero en salir.
- Las pilas se utilizan para manejar invocaciones de funciones (la pila de llamadas), para operaciones como deshacer / rehacer, y para enrutar (recordar páginas que ha visitado y retroceder / avanzar) y mucho más.
- No son una estructura de datos incorporada en JavaScript, pero son relativamente fáciles de implementar
- Insertar y quitar son ambos **O (1)**

Colas

OBJETIVOS

- Definir qué es una cola
- Comprender los casos de uso de una cola.
- Implementar operaciones en una estructura de datos de cola.

¿QUE ES UNA COLA?

Una estructura de datos **FIFO** !

F rimero **I** n **F** rimero **O** ut

Hemos visto esto antes

¡Las colas existen en todas partes! Piensa en la última vez que esperaste en la fila ...

¿Cómo los usamos en la programación?

- Tarea en segundo plano
- Cargando recursos
- Impresión / procesamiento de tareas

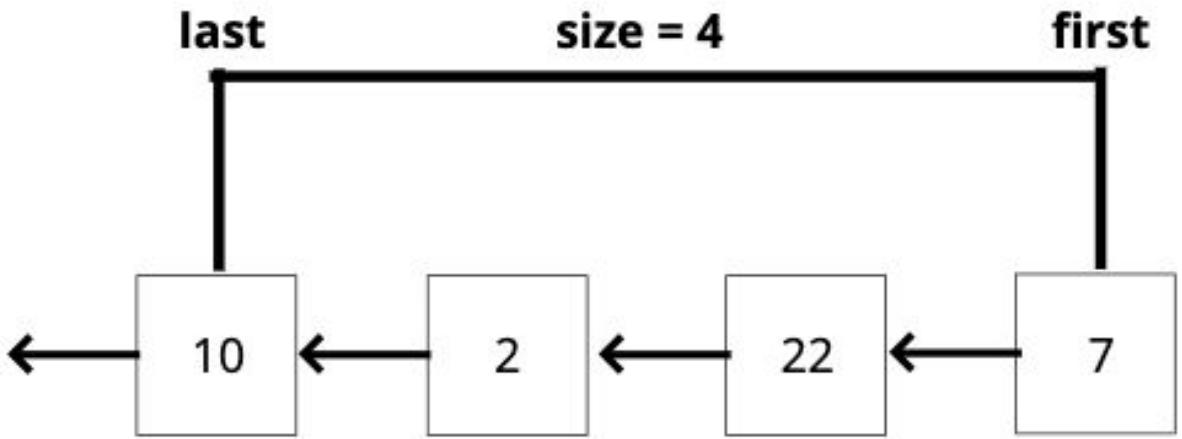
CONSTRUIR UNA COLA CON UN ARRAY

Una clase de cola

```
class Queue {
    constructor(){
        this.first = null;
        this.last = null;
        this.size = 0;
    }
}
class Node {
    constructor(value){
        this.value = value;
        this.next = null;
    }
}
```

Cómo visualizamos una cola

A series of nodes!



Encolar

¡Agregando al **principio** de la cola!

Recuerda, las colas son una estructura de datos **FIFO.**

Encolar Pseudocódigo

- Esta función acepta algún valor.
- Crea un nuevo nodo usando ese valor pasado a la función
- Si no hay nodos en la cola, configure este nodo para que sea la primera y la última propiedad de la cola
- De lo contrario, establezca la siguiente propiedad en la última actual para que sea ese nodo, y luego configure la última propiedad de la cola para que sea ese nodo
- Incrementa el tamaño de la cola en 1.

```
class Node {
    constructor(value){
        this.value = value;
        this.next = null;
    }
}

class Queue {
    constructor(){
        this.first = null;
        this.last = null;
        this.size = 0;
    }
```

```
 enqueue(val){
        var newNode = new
Node(val);
        if(!this.first){
            this.first = newNode;
            this.last = newNode;
        } else {
            this.last.next = newNode;
            this.last = newNode;
        }
        return ++this.size;
    }
```

Desencolar

¡Eliminando desde el **principio** de la Cola!

Recuerda, las colas son una estructura de datos **FIFO.**

Desencolar pseudocódigo

- Si no hay primera propiedad, solo devuelve null
- Almacena la primera propiedad en una variable.
- Vea si el primero es el mismo que el último (verifique si solo hay 1 nodo). Si es así, establece el primero y el último en ser nulo
- Si hay más de 1 nodo, configure la primera propiedad como la siguiente propiedad de la primera
- Disminuir el tamaño en 1
- Devuelve el valor del nodo en cola

```
dequeue(){
        if(!this.first) return null;

        var temp = this.first;
        if(this.first === this.last) {
            this.last = null;
        }
        this.first = this.first.next;
        this.size--;
        return temp.value;
    }
}
```

Gran o de las colas

Inserción - **O (1)**

Eliminación - **O (1)**

Buscando - **O (N)**

Acceso - **O (N)**

RESUMEN

- Las colas son una estructura de datos FIFO , todos los elementos son los primeros en salir primero.
- Las colas son útiles para procesar tareas y son fundamentales para estructuras de datos más complejas
- La inserción y la eliminación se pueden hacer en O (1)