

Estructura de Datos

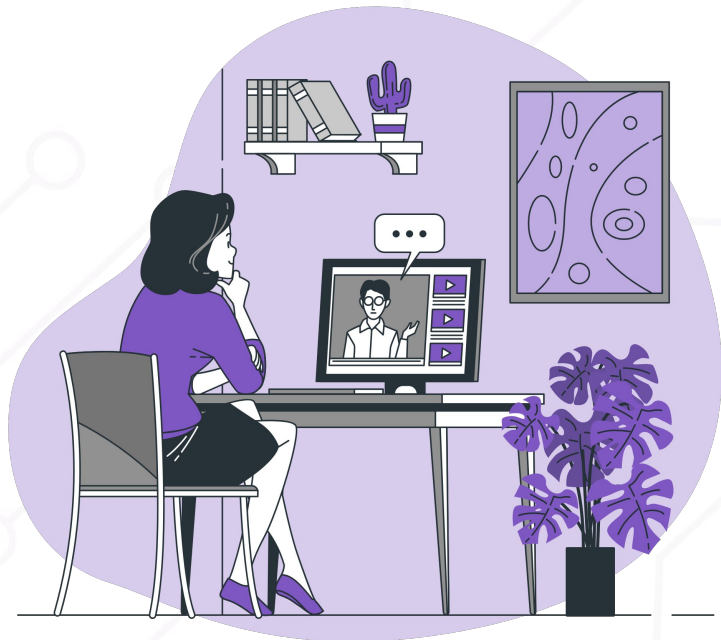
COLAS

DEV.F
DESARROLLAMOS(PERSONAS);

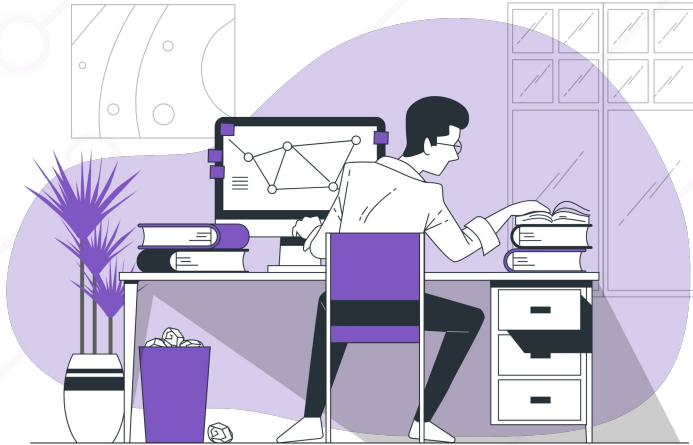
dev

OBJETIVOS DE LA SESIÓN

1. EL ALCANCE DE LA CLASE SERÁ INTRODUCCIÓN A LAS COLAS (QUEUE) EN JAVASCRIPT
2. ENTENDER QUE SON LAS COLAS Y CÓMO FUNCIONAN.
3. USAREMOS UN CÓDIGO DE EJEMPLO PARA VER CÓMO SE PORTAN LAS COLAS.
4. SEGUIREMOS CON LA DINÁMICA DE LOS EJERCICIOS PARA REFORZAR EL TEMA VISTO EN CLASE CON FINES DE PRACTICAR.

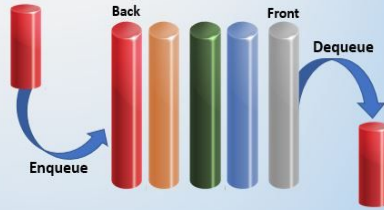


¿Qué son las colas (Queue) ?



Una cola es una estructura de datos muy similar a una **Pila**, es decir, también opera de forma lineal y unidireccional (**se agregan elementos de inicio a fin**).

Queue in JavaScript



www.educba.com

COLAS

Una cola (queue) es una estructura de datos compuesta por una serie de elementos donde insertamos data al final de la serie y retiramos data por el frente

FIFO



FIRST IN



FIRST OUT

F.I.F.O.

- Es lo que llamamos una estructura FIFO (FIRST IN, FIRST OUT). Usamos colas para varias cosas como por ejemplo:

ORDENAR OPERACIONES

COLA DE IMPRESIÓN

Descripción Gráfica

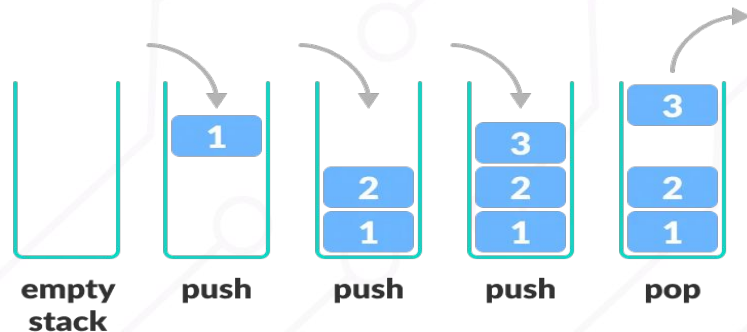


Queue



DATA STRUCTURE BASICS

La gran diferencia radica en la forma en que estos elementos son sacados después. Cuando usamos una **Pila**, ésta opera con una modalidad **LIFO (Last In First Out)**.



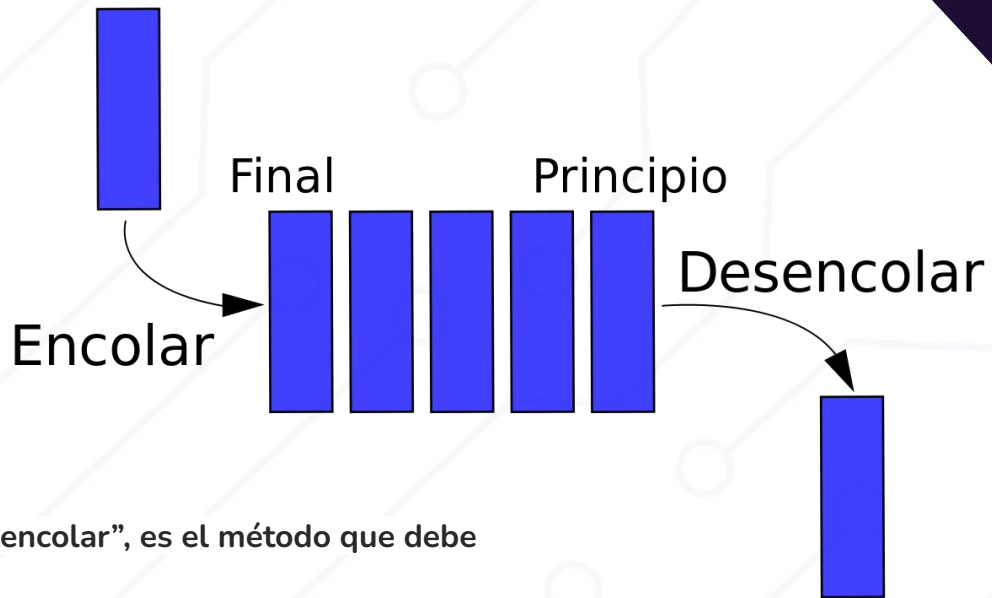
Mientras que con una **Cola** es **FIFO (First In First Out)**, es decir, siempre el primer elemento que agreguemos, será el primero que saquemos de ella.



Colas

En principio las **colas** o **queues** tienen dos métodos importantes:

1. **enqueue** : Traducido como “encolar”, es el método que debe agregar un elemento a la cola.
2. **dequeue** : Traducido como “desencolar”. Este método retira el primer elemento de la cola.



MÉTODOS DE UNA COLA

Para poder operar correctamente sobre una cola, es necesario contar con métodos que permitan: **agregar elementos, sacar elementos, mostrar elementos** (*el siguiente o la cola completa*) y retornar el tamaño.

- **enqueue:** Agrega un nuevo elemento a la cola, **situándolo al final de ésta.**
- **dequeue:** Retorna el primer elemento de la cola, **quitándolo de ésta.**
- **peek:** Retorna el primer elemento de la cola, **sin quitarlo de ésta.**
- **size:** Retorna el número de elementos que contiene la cola.
- **print:** Muestra el contenido de la cola.

MÉTODOS AUXILIARES

Aunque hay otros métodos auxiliares que pueden ser de mucha utilidad:

1. **isEmpty** : indica si la cola está vacía.
2. **front** : nos permite conocer el primer elemento agregado, es decir, el primer elemento de la cola.

En algunos casos, nos permite obtener el valor del primer elemento, pero siempre, sin retirar el elemento de la cola.
3. **back** : nos permite conocer el último elemento de la cola, es decir, el último en ser agregado. En algunos casos, nos permite obtener el valor, pero siempre, sin retirar el elemento de la cola.

Listas Enlazadas

(Linked List)

DEV.F
DESARROLLAMOS(PERSONAS);

dev

TIPOS DE LISTAS ENLAZADAS

- **Singly Linked Lists:** Cada nodo contiene un único puntero hacia el siguiente nodo.
- **Doubly Linked Lists:** Cada nodo contiene dos punteros, uno al siguiente nodo y otro al anterior.
- **Circular Linked Lists:** Variación en donde el nodo final apunta hacia el nodo inicial.

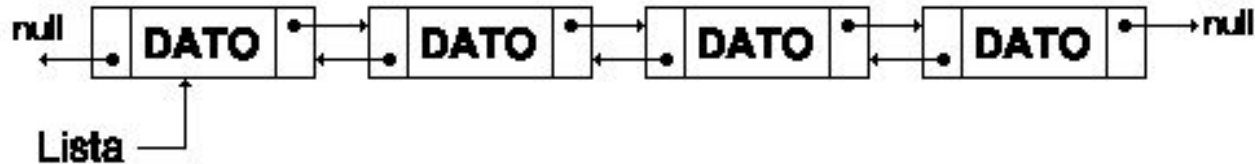
Listas simplemente enlazadas

Se componen de nodos que tienen
Dos atributos
Datos y enlace al siguiente nodo



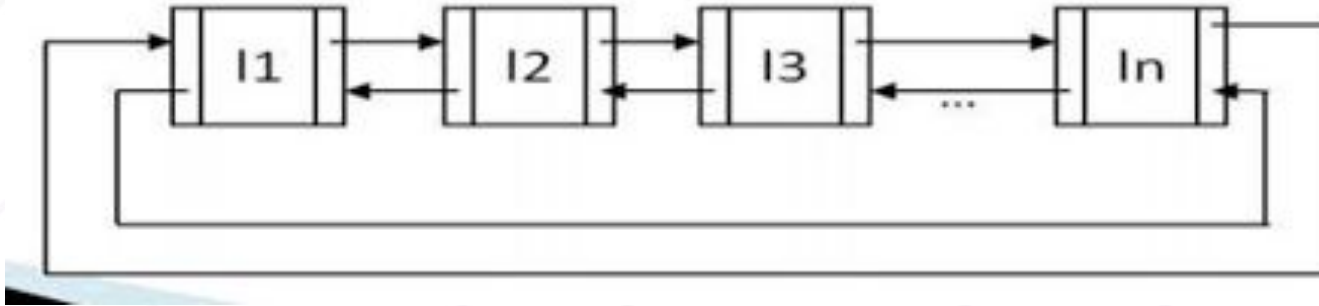
Listas Doblemente enlazadas

Se componen de nodos que tienen
Tres atributos
Datos y enlace al nodo siguiente y al anterior



Listas enlazadas circulares

Se componen de nodos que tienen
Tres atributos
Datos y enlace al nodo siguiente y al anterior
además de conectar los nodos inicial y final



Listas enlazadas

```
const list = {  
  head: {  
    value: 6  
    next: {  
      value: 10  
      next: {  
        value: 12  
        next: {  
          value: 3  
          next: null  
        }  
      }  
    }  
  }  
};
```


Ventajas de las Listas enlazadas

Los nodos pueden ser fácilmente removidos o añadidos a la lista enlazada sin tener que reorganizar toda la estructura. Esta es la principal ventaja frente a los Arrays.

Desventajas de las Listas enlazadas

- Las operaciones de búsqueda son más lentas y costosas, puesto que no se puede acceder a un nodo aleatorio, el recorrido debe hacerse de manera secuencial.
- Utiliza más memoria que un Array puesto que también debe almacenar el puntero.

MÉTODOS LISTAS ENLAZADAS

- **add:** añade un elemento a la lista.
- **remove:** elimina un elemento de la lista.
- **size:** retorna el tamaño de la lista.
- **print:** nos muestra el contenido de la lista.
- **isEmpty:** nos retorna un booleano indicando si la lista está o no vacía.
- **clear:** elimina todos los elementos de la lista.
- **getFirst:** nos retorna el primer elemento de la lista.
- **getLast:** nos retorna el último elemento de la lista.
- **search:** nos retorna el nodo que concuerde con la búsqueda.