Estructura de Datos

Listas Enlazadas

DEV.F:
DESARROLLAMOS(PERSONAS);

OBJETIVOS DE LA SESIÓN

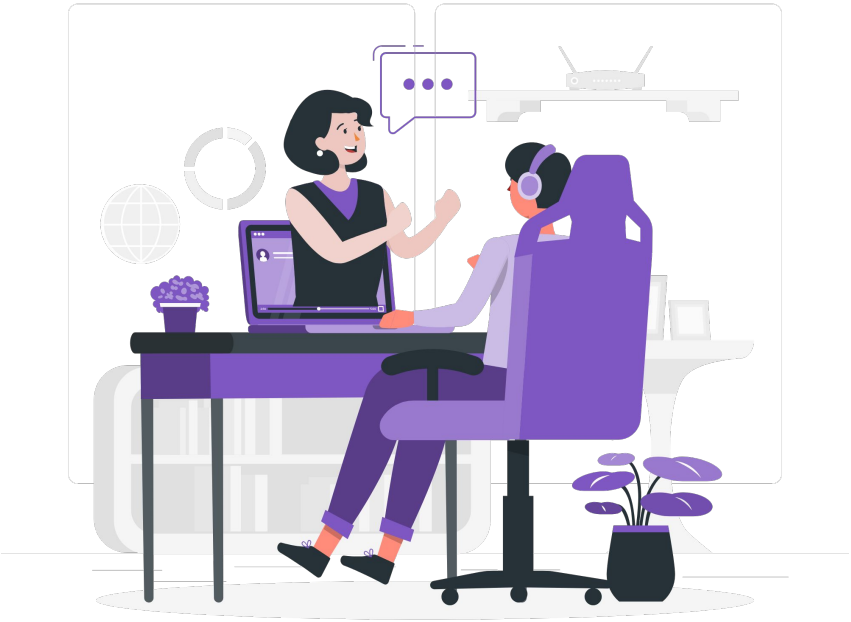
1. EL ALCANCE DE LA CLASE SERÁ INTRODUCCIÓN A LAS LISTAS ENLAZADAS (Linked List) EN JAVASCRIPT

2. ENTENDER QUE SON LAS LISTAS ENLAZADAS CÓMO FUNCIONAN,LAS OPERACIONES QUE PODEMOS HACER CON ELLAS .

3. HAREMOS CÓDIGO PARA VER SU COMPORTAMIENTO.

4. SEGUIREMOS CON LA DINÁMICA DE LOS EJERCICIOS PARA REFORZAR EL TEMA VISTO EN CLASE CON FINES DE PRACTICAR.

DEV.F



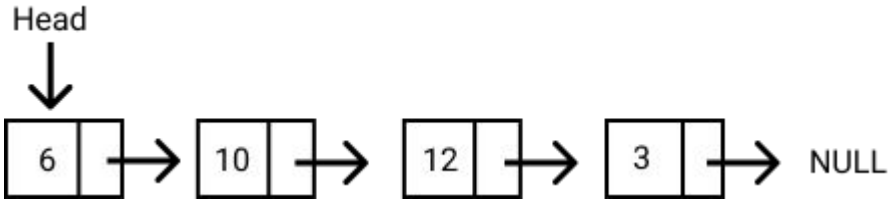
Listas Enlazadas

Una lista enlazada es una estructura de datos lineal similar a un arreglo. Sin embargo, a diferencia de los arreglos, los elementos no se almacenan de manera contigua. Cada elemento es un objeto independiente que contiene un puntero o un enlace al siguiente objeto de la lista.

DEV.F::

Listas Enlazadas

El punto de entrada a una lista enlazada se llama cabeza (head). La cabeza es una referencia al primer nodo de la lista enlazada.



El último nodo de la lista apunta a null. Si una lista está vacía, la cabeza es una referencia nula.

Cada elemento (comúnmente llamados nodos) contiene dos elementos:

1. Los datos almacenados y
2. Un enlace al siguiente nodo.

Los datos pueden ser de cualquier tipo válido.

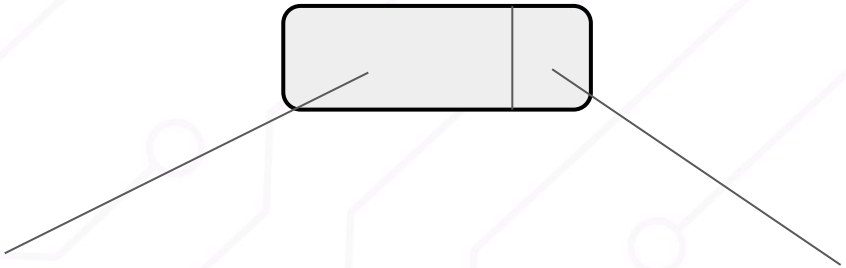
DEV.F::

En código

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

Representación Gráfica.

Las listas enlazadas permiten almacenar información en posiciones de memoria que no sean contiguas; para almacenar la información contienen elementos llamados nodos estos nodos poseen dos campos:



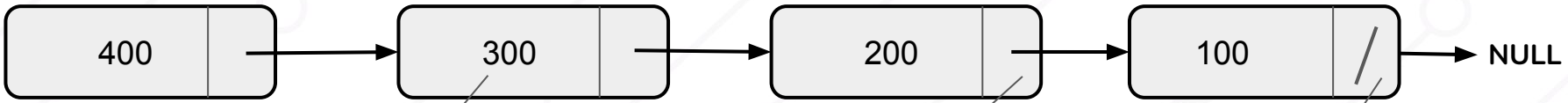
uno para almacenar la información o valor del elemento (data).

otro para el enlace que determina la posición del siguiente elemento o nodo de la lista.

Anatomía de las listas enlazadas

Representación
De un nodo:

Head
(primer nodo)

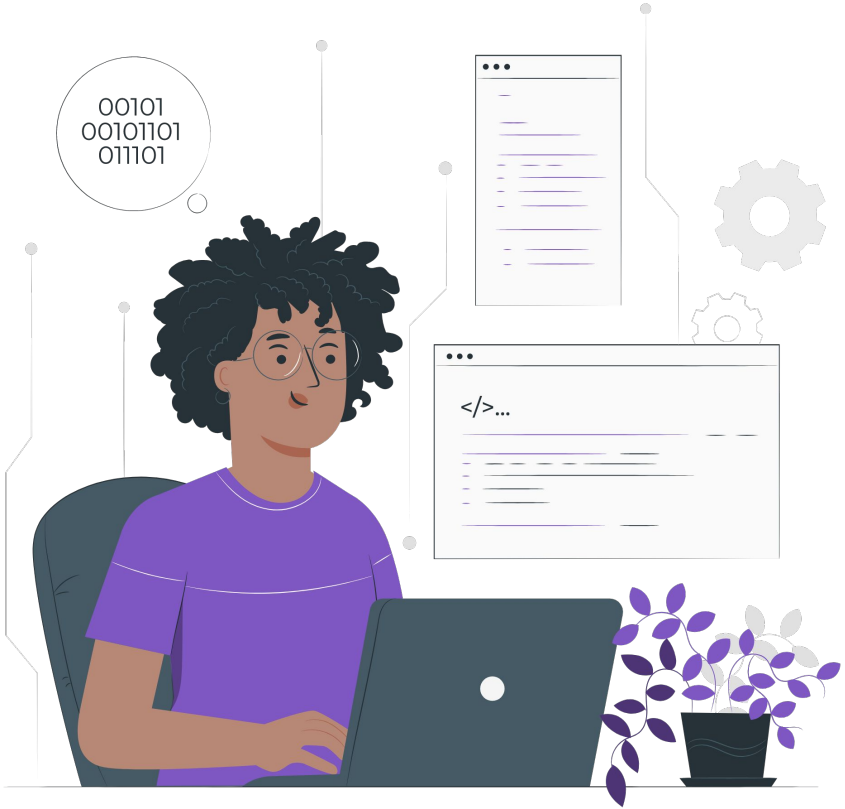


Cada nodo almacena un dato, en este caso números.

Cada nodo tiene un sucesor:
Un nodo siguiente

El último nodo no tiene sucesor:
Su puntero apunta a null

DEV.F



VENTAJAS DE LAS LISTAS ENLAZADAS

Los nodos pueden eliminarse o añadirse fácilmente de una lista enlazada sin reorganizar toda la estructura de datos. Esta es una ventaja que tiene sobre los arreglos

DEV.F::

Implementación de un nodo en JavaScript

Como ya se ha dicho, un nodo de lista contiene dos elementos: los datos y el puntero al siguiente nodo.

Podemos implementar un nodo en JavaScript de la siguiente manera:

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

Implementación de una lista enlazada en JavaScript

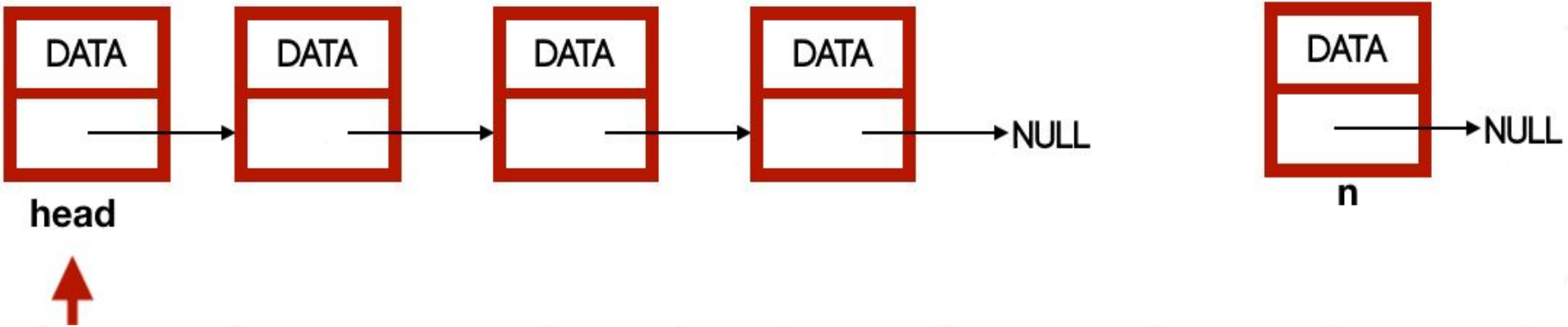
El código siguiente muestra la implementación de una clase de lista enlazada con un constructor. Observe que si no se pasa el nodo cabeza, la cabeza se inicializa a null.

```
class LinkedList {
    constructor(head = null) {
        this.head = head;
    }

}
```

```
class LinkedList {
    constructor() {
        this.head = null;
    }

}
```


DATA
DATA
DATA
DATA
NULL
head
DATA
NULL
n

Como tip

Lo más recomendable y flexible para la creación de un nodo es utilizar un objeto por cada nodo.

Operaciones que podemos hacer con una lista

- Inserción de un elemento.
- Borrado de un elemento.
- Recorrido de la lista.
- Búsqueda de un elemento.