



# Universidad Tecnológica de Metropolitana

**Estructura de datos | Parcial 2. Listas**

**Profesor: Ruth Betsaida Martinez Dominguez**

**Alumno: Abril Contreras Suaste**

**Tarea: Actividades del 5 al 8**

**10 de Octubre de 2024**

## Ejercicio 5. Lista de palabras ordenadas

### Explicación del código

1. Clase NodoPalabra Representa cada palabra en la lista enlazada de palabras. Cada nodo contiene una palabra y una referencia al siguiente nodo en la lista.

```
class NodoPalabra {  
    constructor(palabra, siguiente = null) {  
        this.palabra = palabra; // Palabra que se almacena  
        this.siguiente = siguiente; // Referencia al siguiente nodo  
    }  
}
```

2. Clase NodoLetra Representa una lista de palabras que comienzan con una misma letra. Cada nodo tiene una referencia a su lista de palabras y siguiente nodo de letra.

```
class NodoLetra {  
    constructor(letra, siguiente = null) {  
        this.letra = letra;  
        this.siguiente = siguiente;  
        this.listaPalabras = null;  
    }  
}
```

3. Clase ListaLetras Es la lista principal que contiene nodos de letra. Tiene métodos para agregar palabras a la lista, crear nuevas listas para letras cuando es necesario, y recorrer todas las letras y sus palabras.

## Ejercicio 6. Invertir una palabra con lista enlazada

### Explicación del código

1. Clase Nodo Define la estructura de un nodo. Cada nodo contiene un carácter y una referencia al siguiente nodo

```
class Nodo {  
  constructor(caracter, siguiente = null) {  
    this.caracter = caracter;  
    this.siguiente = siguiente;  
  }  
}
```

2. Clase ListaEnlazada Esta clase es la responsable de manejar la lista de nodos. Tiene los siguientes métodos: 1. Método agregarCaracter(caracter): Inserta un nuevo nodo con el carácter al final de la lista

```
  agregarCaracter(caracter) {  
    const nuevoNodo = new Nodo(caracter);  
    if (!this.head) {  
      this.head = nuevoNodo;  
    } else {  
      let actual = this.head;  
      while (actual.siguiente) {  
        actual = actual.siguiente;  
      }  
      actual.siguiente = nuevoNodo;  
    }  
  }  
}
```

2. Método invertirLista(): Invierte el orden de los nodos en la lista enlazada, cambiando la dirección de las referencias siguiente

```
invertirLista() {  
  let anterior = null;  
  let actual = this.head;  
  let siguiente = null;  
  while (actual !== null) {  
    siguiente = actual.siguiente;  
    actual.siguiente = anterior;  
    anterior = actual;  
    actual = siguiente;  
  }  
  this.head = anterior;  
}
```

3. Función invertirPalabra(): Toma la palabra que el usuario ingresa, descomone cada carácter y los inserta en una lista enlazada mediante

el método `agregarCaracter()`. Despuews, la lista enlazada es invertida mediante el método `invertirLista()` y finalmente, se recorre la lista invertida para construir la palabra invertida.

```
function invertirPalabra() {  
  const palabra = document.getElementById('palabra').value.trim();  
  if (palabra) {  
    const lista = new ListaEnlazada();  
    for (let i = 0; i < palabra.length; i++) {  
      lista.agregarCaracter(palabra[i]);  
    }  
  
    // Invertir la lista enlazada  
    lista.invertirLista();  
  }  
}
```

## Ejercicio 7. Verificar palíndromo

### Explicación del código

1. Clase Nodo: Define la estructura de un nodo. Cada nodo contiene un carácter y una referencia al siguiente nodo.
2. Clase ListaEnlazada: Está encargada de manejar la lista de nodos. Tiene los siguientes métodos:
  1. Método agregarCaracter(caracter): Inserta un nuevo nodo con el carácter al final de la lista.
  2. invertirLista(): Invierte el orden de los nodos en la lista enlazada.
  3. obtenerPalabra(): Convierte la lista enlazada en una cadena de texto.
3. verificarPalidromo(): Toma la palabra que el usuario ingresa, descompone cada carácter y los inserta en una lista enlazada mediante el método agregarCaracter(). Después, la lista enlazada es invertida mediante el método invertirLista() y se compara con la palabra original para verificar si es un palíndromo.

```
function verificarPalindromo() {  
    const fraseOriginal = document.getElementById('palabra').value.trim();  
    const palabraLimpia = limpiarTexto(fraseOriginal);  
  
    if (palabraLimpia) {  
        const listaOriginal = new ListaEnlazada();  
        const listaInvertida = new ListaEnlazada();  
  
        for (let i = 0; i < palabraLimpia.length; i++) {  
            listaOriginal.agregarCaracter(palabraLimpia[i]);  
            listaInvertida.agregarCaracter(palabraLimpia[i]);  
        }  
  
        listaInvertida.invertirLista();  
    }  
}
```