



Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón

Ingeniería en Computación

Asignatura: Estructura de datos

TAREA 4: Implementación de la estructura de datos Lista Ligada

Profesor: Jesús Hernández Cabrera

Alumno: Juan Diego Ortiz Cruz

Grupo: 1360

Fecha: 03/09/2024

Captura Clase Nodo:



```
1  export class Nodo<T> {
2      private dato: T;
3      private siguiente: Nodo<T> | null;
4
5      constructor();
6      constructor(dato: T);
7      constructor(dato: T, siguiente: Nodo<T>);
8      constructor(dato?: T, siguiente?: Nodo<T>) {
9          this.dato = dato as T;
10         this.siguiente = siguiente || null;
11     }
12
13     public getDato(): T {
14         return this.dato;
15     }
16
17     public setDato(dato: T): void {
18         this.dato = dato;
19     }
20
21     public getSiguiente(): Nodo<T> | null {
22         return this.siguiente;
23     }
24
25     public setSiguiente(siguiente: Nodo<T> | null): void {
26         this.siguiente = siguiente;
27     }
28
29     public toString(): string {
30         return `Nodo{dato=${this.dato}, siguiente=${this.siguiente}}`;
31     }
32 }
```

Capturas de la clase Lista Ligada:

```
1  import { Nodo } from './nodo';
2
3  export class ListaLigada<T> {
4      private head: Nodo<T> | null;
5      private tamano: number;
6
7      constructor() {
8          this.head = null;
9          this.tamano = 0;
10     }
11
12     /**
13      * estaVacia
14      */
15     public estaVacia(): boolean {
16         return this.head === null;
17     }
18
19     /**
20      * getTamano
21      */
22     public getTamano(): number {
23         return this.tamano;
24     }
25
26     /**
27      * agregarAlInicio
28      */
29     public agregarAlInicio(dato: T): void {
30         this.head = new Nodo<T>(dato, this.head!);
31         this.tamano++;
32     }
33 }
```

```
1
2 /**
3  * agregarAlFinal
4  */
5 public agregarAlFinal(dato: T): void {
6     const nuevoNode = new Node<T>(dato);
7     if (this.head === null) {
8         this.head = nuevoNode;
9     } else {
10         let aux = this.head;
11         while (aux.getSiguiente() !== null) {
12             aux = aux.getSiguiente();
13         }
14         aux.setSiguiente(nuevoNode);
15     }
16     this.tamano++;
17 }
18
19 public agregarDespuesDe(referencia: T, valor: T): void {
20     let aux = this.head;
21     while (aux !== null && aux.getDato() !== referencia) {
22         aux = aux.getSiguiente();
23     }
24     if (aux !== null) {
25         const nuevoNode = new Node<T>(valor, aux.getSiguiente());
26         aux.setSiguiente(nuevoNode);
27         this.tamano++;
28     }
29 }
30
31 public eliminarElPrimero(): void {
32     if (this.head !== null) {
33         this.head = this.head.getSiguiente();
34         this.tamano--;
35     }
36 }
37
38 public eliminarElFinal(): void {
39     if (this.head === null) return;
40     let aux = this.head;
41     while (aux.getSiguiente()?.getSiguiente() !== null) {
42         aux = aux.getSiguiente();
43     }
44     aux.setSiguiente(null);
45     this.tamano--;
46 }
```

```
1 public eliminar(posicion: number): void {
2     if (posicion < 0 || posicion >= this.tamano) {
3         throw new Error("Posición fuera de rango");
4     }
5     if (posicion === 0) {
6         this.eliminarElPrimero();
7     } else {
8         let aux = this.head!;
9         for (let i = 0; i < posicion - 1; i++) {
10             aux = aux.getSiguiente()!;
11         }
12         aux.setSiguiente(aux.getSiguiente()!.getSiguiente());
13         this.tamano--;
14     }
15 }
16
17 public buscar(valor: T): number {
18     let aux = this.head;
19     let posicion = 0;
20     while (aux !== null) {
21         if (JSON.stringify(aux.getDato()) === JSON.stringify(valor)) {
22             return posicion;
23         }
24         aux = aux.getSiguiente()!;
25         posicion++;
26     }
27     return -1;
28 }
29
30 public actualizar(aBuscar: T, valor: T): void {
31     let aux = this.head;
32     while (aux !== null) {
33         if (JSON.stringify(aux.getDato()) === JSON.stringify(aBuscar)) {
34             aux.setDato(valor);
35             return;
36         }
37         aux = aux.getSiguiente()!;
38     }
39 }
40
41 /**
42  * transversal
43  */
44 public transversal(): void {
45     let aux = this.head;
46     process.stdout.write("|");
47     while (aux !== null) {
48         process.stdout.write(`${aux.getDato()}| -> |`);
49         aux = aux.getSiguiente()!;
50     }
51     console.log("null|");
52 }
53 }
```

Captura de la clase SmartPhone:

```
1  export class SmartPhone {
2      private marca: string;
3      private modelo: string;
4      private precio: number;
5
6      constructor(marca: string, modelo: string, precio: number) {
7          this.marca = marca;
8          this.modelo = modelo;
9          this.precio = precio;
10     }
11
12     public getMarca(): string {
13         return this.marca;
14     }
15
16     public getModelo(): string {
17         return this.modelo;
18     }
19
20     public getPrecio(): number {
21         return this.precio;
22     }
23
24     public setPrecio(precio: number): void {
25         this.precio = precio;
26     }
27
28     public toString(): string {
29         return `${this.marca} ${this.modelo} (${this.precio})`;
30     }
31 }
32
```

Capturas de la clase Main:

```
1 import { ListaLigada } from './ListaLigada';
2 import { SmartPhone } from './SmartPhone';
3
4 class Main {
5     public static main(): void {
6         // Crear una lista de SmartPhones
7         const listaSmartPhones = new ListaLigada<SmartPhone>();
8
9         // Agregar 5 SmartPhones
10        listaSmartPhones.agregarAlFinal(new SmartPhone("Apple", "iPhone 12", 999));
11        listaSmartPhones.agregarAlFinal(new SmartPhone("Samsung", "Galaxy S21", 899));
12        listaSmartPhones.agregarAlFinal(new SmartPhone("Google", "Pixel 5", 699));
13        listaSmartPhones.agregarAlFinal(new SmartPhone("OnePlus", "9 Pro", 969));
14        listaSmartPhones.agregarAlFinal(new SmartPhone("Xiaomi", "Mi 11", 749));
15
16        // Imprimir el contenido
17        console.log("Lista original:");
18        listaSmartPhones.transversal();
19
20        // Eliminar el de la posición 2
21        listaSmartPhones.eliminar(2);
22
23        // Volver a imprimir el contenido
24        console.log("\nLista después de eliminar el elemento en la posición 2:");
25        listaSmartPhones.transversal();
26
27        // Actualizar el segundo elemento
28        const aBuscar = new SmartPhone("Samsung", "Galaxy S21", 899);
29        const nuevoSmartPhone = new SmartPhone("Samsung", "Galaxy S21 Ultra", 1199);
30        listaSmartPhones.actualizar(aBuscar, nuevoSmartPhone);
31
32        // Volver a imprimir el contenido
33        console.log("\nLista después de actualizar el segundo elemento:");
34        listaSmartPhones.transversal();
35
36        // Agregar un elemento al inicio y otro diferente al final
37        listaSmartPhones.agregarAlInicio(new SmartPhone("Huawei", "P40 Pro", 799));
38        listaSmartPhones.agregarAlFinal(new SmartPhone("Sony", "Xperia 1 III", 1299));
39
40        // Volver a imprimir el contenido
41        console.log("\nLista después de agregar al inicio y al final:");
42        listaSmartPhones.transversal();
43
44        // Eliminar el primero
45        listaSmartPhones.eliminarElPrimero();
46
47        // Volver a imprimir el contenido
48        console.log("\nLista final después de eliminar el primer elemento:");
49        listaSmartPhones.transversal();
50    }
51 }
52
53 // Ejecutar el método main
54 Main.main();
```

Capturas consola (Ejecución):

```
PS C:\Users\LIMA\Desktop\Proyectos\Estructura-de-datos\tarea4> tsx main.ts
```

```
Lista original:
```

```
|Apple iPhone 12 ($999)| -> |Samsung Galaxy S21 ($899)| -> |Google Pixel 5 ($699)| -> |OnePlus 9 Pro ($969)| -> |Xiaomi Mi 11 ($749)| -> |null|
```

```
Lista después de eliminar el elemento en la posición 2:
```

```
|Apple iPhone 12 ($999)| -> |Samsung Galaxy S21 ($899)| -> |OnePlus 9 Pro ($969)| -> |Xiaomi Mi 11 ($749)| -> |null|
```

```
Lista después de actualizar el segundo elemento:
```

```
|Apple iPhone 12 ($999)| -> |Samsung Galaxy S21 Ultra ($1199)| -> |OnePlus 9 Pro ($969)| -> |Xiaomi Mi 11 ($749)| -> |null|
```

```
Lista después de agregar al inicio y al final:
```

```
|Huawei P40 Pro ($799)| -> |Apple iPhone 12 ($999)| -> |Samsung Galaxy S21 Ultra ($1199)| -> |OnePlus 9 Pro ($969)| -> |Xiaomi Mi 11 ($749)| -> |Sony Xperia 1 III ($1299)| -> |null|
```

```
Lista final después de eliminar el primer elemento:
```

```
|Apple iPhone 12 ($999)| -> |Samsung Galaxy S21 Ultra ($1199)| -> |OnePlus 9 Pro ($969)| -> |Xiaomi Mi 11 ($749)| -> |Sony Xperia 1 III ($1299)| -> |null|
```