



Ingeniería en Software

INFORME ESTRUCTURA DE DATOS

<CaracolaMagic>

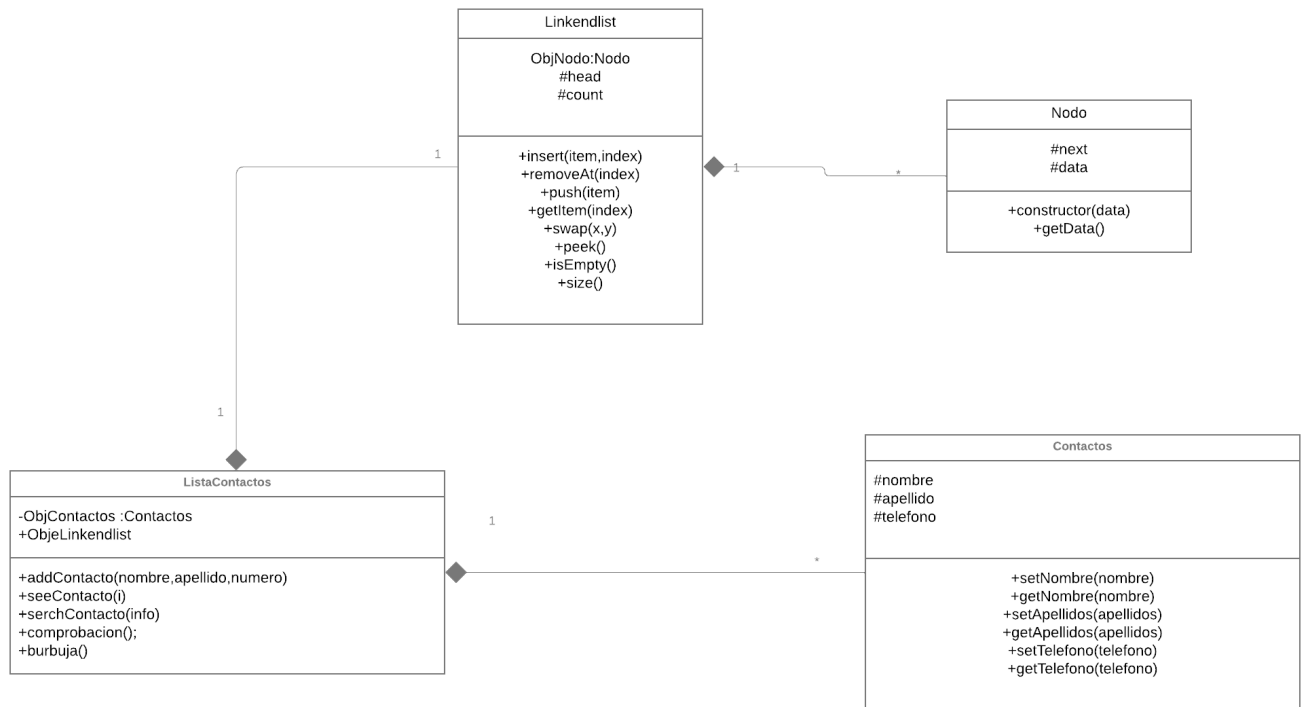
<Christian Alejandro Torres Hernandez>

<Hector Roman Robles Orantes>

Suchiapa, 2024

Fase de análisis

Diagrama de clases



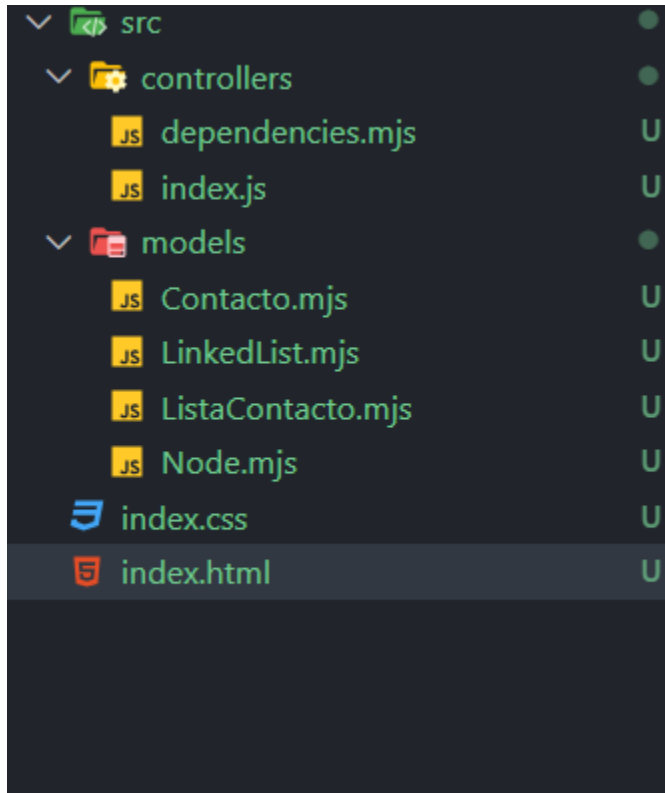
Repositorio del proyecto

<https://github.com/Orantes67/CaracolaMagic.git>

<https://orantes67.github.io/CaracolaMagic/>

Fase de desarrollo

Estructura de carpetas



Código

```
export class Node {
  #data
  #next
  constructor(data){
    this.#data = data
    this.#next = null
  }
  getData () {
    return this.#data
  }
}
```

```
import {Contacto} from "../Contacto.mjs"
import { linkedlist } from "../controllers/dependencies.mjs";
export class listaContacto{
  addContacto(nombre,apellido,numero){
    let contacto = new Contacto()
    contacto.setNombre(nombre)
    contacto.setApellido(apellido)
    contacto.setTelefono(numero)

    linkedlist.push(contacto)
  }
  seeContacto(i){
    let objcontacto= linkedlist.getItem(i)
    return objcontacto
  }
  searchContacto(info) {
    for (let i = 0; i < linkedlist.size(); i++) {
      let contacto = linkedlist.getItem(i);
      if (contacto.getNombre() == info) {
        return contacto;
      }
    }
    return null;
  }
  comprobacion(){
    return linkedlist.size()
  }
  burbuja(){
    let n, i, k;
    n = linkedlist.size();
    let aux = new Contacto()
    for (k = 1; k <= n; k++) {
      for (i = 0; i <= (n - k); i++) {
        console.log(linkedlist.getItem(i).getNombre())
        console.log(linkedlist.getItem(i+1).getNombre())
        console.log(linkedlist.getItem(i).getNombre())
        >linkedlist.getItem(i+1).getNombre()
        if (linkedlist.getItem(i).getNombre()>linkedlist.getItem(i+1).getNombre())
        {
          aux.setNombre(linkedlist.getItem(i).getNombre())
          aux.setApellido(linkedlist.getItem(i).getApellido())
          aux.setTelefono(linkedlist.getItem(i).getTelefono())

          linkedlist.getItem(i).setNombre(linkedlist.getItem(i+1).getNombre())
          linkedlist.getItem(i).setApellido(linkedlist.getItem(i+1).getApellido())
          linkedlist.getItem(i).setTelefono(linkedlist.getItem(i+1).getTelefono())

          linkedlist.getItem(i+1).setNombre( aux.getNombre())
          linkedlist.getItem(i+1).setApellido( aux.getApellido())
          linkedlist.getItem(i+1).setTelefono( aux.getTelefono())
        }
      }
    }
  }
}
```

```

import { Node } from "../Node.mjs"
export class LinkedList {
  #count
  #head
  constructor() {
    this.#head = null;
    this.#count = 0;
  }

  insert(item, index) {
    const node = new Node(item)
    if (index === 0) {
      node.next = this.#head;
      this.#head = node;
    } else {
      let current = this.#head;
      let currentPosition = 0;
      while (currentPosition < index - 1 && current !== null && current.next !== null) {
        current = current.next;
        currentPosition++;
      }
      if (current !== null) {
        if (current.next !== null) {
          node.next = current.next;
        }
        current.next = node;
      } else {
        //
      }
    }
  }

  removeAt(index) {
    if (index === 0) {
      return null;
    }
    let removedNode = null;
    if (index === 0) {
      if (this.#head !== null) {
        removedNode = this.#head;
        this.#head = this.#head.next;
        return removedNode;
      } else {
        return null;
      }
    }
    let current = this.#head;
    let currentIndex = 0;
    while (current !== null && currentIndex < index - 1) {
      current = current.next;
      currentIndex++;
    }
    if (current === null || current.next === null) {
      return null;
    }
    removedNode = current.next;
    current.next = current.next.next;
    return removedNode;
  }

  push(item) {
    const node = new Node(item)
    if (!this.#head) {
      this.#head = node;
    } else {
      let current = this.#head;
      while (current.next) {
        current = current.next;
      }
      current.next = node;
    }
    this.#count++
  }

  getItem(index) {
    if (index < 0 || index >= this.size()) {
      console.log("Out of range");
      return -1;
    }
    let currentIndex = 0;
    let currentNode = this.#head;
    while (currentIndex < index) {
      currentNode = currentNode.next;
      currentIndex++;
    }
    return currentNode.getData();
  }


  swap(x, y) {
    let current = this.#head;
    if (!current) return;
    if (current.getData() === x) {
      current.data = y;
    } else if (current.getData() === y) {
      current.data = x;
    }
    current = current.next;
  }

  peek() {
    if (!this.isEmpty()) {
      return this.#head;
    } else {
      return null;
    }
  }

  isEmpty() {
    return this.#head === null;
  }

  size() {
    return this.#count;
  }
}

```



```
export class Contacto {  
  #nombre  
  #apellido  
  #telefono  
  setNombre(nombre){this.#nombre = nombre}  
  getNombre(){return this.#nombre}  
  setApellido(apellido){this.#apellido = apellido}  
  getApellido(){return this.#apellido}  
  setTelefono(telefono){this.#telefono = telefono}  
  getTelefono(){return this.#telefono}  
}
```

```

import { listContacts } from "../dependencies.mjs";

console.log("a-c-z");
let addcontac = document.getElementById("btn-add");
addcontac.addEventListener("click", function() {
  const card = document.createElement("div");
  divchar.appendChild(card)

  let pName = document.createElement("p");
  pName.innerText = "ingrese el nombre del contacto";
  card.appendChild(pName);
  let charName = document.createElement("input");
  charName.setAttribute("id", "inputName");

  let pLName = document.createElement("p");
  pLName.innerText = "ingrese el apellido del contacto";
  card.appendChild(pLName);
  let charLName = document.createElement("input");
  card.appendChild(charLName);
  charLName.setAttribute("id", "inputLName");

  let pCelular = document.createElement("p");
  pCelular.innerText = "ingrese el número telefónico del contacto";
  card.appendChild(pCelular);
  let charCelular = document.createElement("input");
  card.appendChild(charCelular);
  charCelular.setAttribute("id", "inputCelular");
  charCelular.setAttribute("type", "number");

  let Btn = document.createElement("button");
  Btn.innerText = "agregar contacto";
  card.appendChild(Btn);

  Btn.addEventListener("click", function() {
    let infoName = null;
    let infoLName = null;
    let infoCelular = null;

    infoName = document.getElementById("inputName").value;
    infoLName = document.getElementById("inputLName").value;
    infoCelular = document.getElementById("inputCelular").value;

    if (infoName != null && infoLName != null && infoCelular != null) {
      listContacts.addContacto({ infoName, infoLName, infoCelular });
      console.log(listContacts.comprobacion());
      infoName = document.getElementById("inputName").value = "";
      infoLName = document.getElementById("inputLName").value = "";
      infoCelular = document.getElementById("inputCelular").value = "";
    } else {
      alert("Por favor, complete todos los campos.");
    }
  });
});

let seecontac = document.getElementById("btn-see");
seecontac.addEventListener("click", function() {
  if (listContacts.comprobacion() > 0) {
    for (let i = 0; i < listContacts.comprobacion(); i++) {
      let contactos = listContacts.seeContacto(i);
      const card = document.createElement("div");
      divchar.appendChild(card);

      let charTXT = document.createElement("h1");
      charTXT.innerText = "nombre";
      card.appendChild(charTXT);
      let charXT = document.createElement("p");
      charXT.innerText = contactos.getNombre();
      card.appendChild(charXT);

      charTXT = document.createElement("h1");
      charTXT.innerText = "apellido";
      card.appendChild(charTXT);
      let charXT = document.createElement("p");
      charXT.innerText = contactos.getApellido();
      card.appendChild(charXT);

      charTXT = document.createElement("h1");
      charTXT.innerText = "numero";
      card.appendChild(charTXT);
      let charXT = document.createElement("p");
      charXT.innerText = contactos.getTelefono();
      card.appendChild(charXT);
    }
  } else {
    alert("Ingrese un contacto.");
  }
});

let findcontac = document.getElementById("btn-find");
findcontac.addEventListener("click", function() {
  if (listContacts.comprobacion() > 0) {
    const card = document.createElement("div");
    divchar.appendChild(card);

    let charTXT = document.createElement("p");
    charTXT.innerText = "ingrese el nombre del contacto";
    card.appendChild(charTXT);
    let char = document.createElement("input");
    card.appendChild(char);
    char.setAttribute("id", "idinput");

    let Btn = document.createElement("button");
    Btn.innerText = "buscar contacto";
    card.appendChild(Btn);
    Btn.addEventListener("click", function() {
      let info = document.getElementById("idinput").value;
      let contac = listContacts.searchContacto(info);

      let charIXI = document.createElement("h1");
      charIXI.innerText = "nombre";
      card.appendChild(charIXI);
      let charXT = document.createElement("p");
      charXT.innerText = contac.getNombre();
      card.appendChild(charXT);

      charTXT = document.createElement("h1");
      charTXT.innerText = "apellido";
      card.appendChild(charTXT);
      let charXT = document.createElement("p");
      charXT.innerText = contac.getApellido();
      card.appendChild(charXT);

      charTXT = document.createElement("h1");
      charTXT.innerText = "numero";
      card.appendChild(charTXT);
      let charXT = document.createElement("p");
      charXT.innerText = contac.getTelefono();
      card.appendChild(charXT);
    });
  } else {
    alert("Ingrese un contacto.");
  }
});

let delcontac = document.getElementById("btn-del");
delcontac.addEventListener("click", function() {
  listContacts.burbuja();
});

```

```

@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  position: relative;
  font-family: 'Poppins', sans-serif;
}

.fondo {
  max-width: 600px;
  margin: 20px auto;
  padding: 20px;
  background-color: #476bc7;
  border-radius: 10px;
  box-shadow: 0 0 10px black;
}

button {
  padding: 10px 20px;
  margin-right: 10px;
  background-color: #244da3;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #0056b3;
}

.card {
  margin-top: 20px;
  padding: 15px;
  border: 1px solid #ccc;
  border-radius: 10px;
  background-color: #f9f9f9;
}

.card h1 {
  font-size: 10px;
  margin-bottom: 10px;
  color: #333;
}

.card p {
  margin: 0;
  color: #666;
}

.form-container {
  display: flex;
  flex-direction: column;
  align-items: flex-start;
}

input {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

.add-contact-btn {
  align-self: flex-end;
  background-color: #28a745;
}

.see-list-btn {
  background-color: #17a2b8;
}

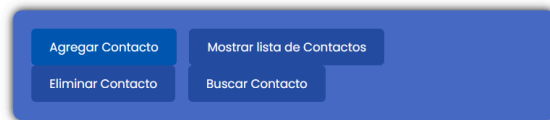
```



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="index.css">
  <title>Queue</title>
</head>
<body>
  <div class="fondo">
    <div>
      <button id="btn-add">Agregar Contacto</button>
      <button id="btn-see">Mostrar lista de Contactos</button>
      <button id="btn-del">ordenar Contactos</button>
      <button id="btn-find">Buscar Contacto</button>
    </div>
    <div id="divchar"></div>
  </div>
  <script type="module" src="./src/controllers/index.js"></script>
</body>
</html>
```

Resultados

Interfaz visual



Agregar Contacto Mostrar lista de Contactos
Eliminar Contacto Buscar Contacto



Agregar Contacto Mostrar lista de Contactos
Eliminar Contacto Buscar Contacto

ingrese el nombre del contacto

ingrese el apellido del contacto

ingrese el numero telefonico del contacto

ingrese el correo electronico del contacto

agregar contacto

agregar contacto

nombre

Ana

apellido

1

numero

1

correo

a

nombre

Ana Maria

apellido

1

numero

1

correo

a

nombre

Hector

apellido

ingrese el nombre del contacto

buscar contacto

ingrese el nombre del contacto

Hector

buscar contacto

nombre

Hector

apellido

1

numero

1

correo

a

Conclusiones

Se mejoró la lógica en la implementación del ordenamiento Burbuja.