GRUPO DE ESTUDOS 8
ESTRUTURA DE DADOS

pilhas
filas
listas

https://github.com/jcbombardelli/grupo-estudo-8-gama-academy


sql.exec(...)
repository.find(...)
service.list(...)
app.get(...)
main( )
Aplicação


app.get(...)
main()
Aplicação
sql.exec(...)
repository.find(...)
service.list(...)


sql.exec(...)
repository.find(...)
service.list(...)
app.get(...)
main( )
Aplicação

```javascript
class Stack {

    constructor() {
        this.items = [];
    }

    push(element) {
        this.items.push(element);
    }

    pop() {
        if (this.items.length === 0) {
            return "Underflow";
        }
        return this.items.pop();
    }

    top() {
        return this.items[this.items.length - 1];
    }

    _isEmpty() {
        return this.items.length === 0;
    }

    size() {
        return this.items.length;
    }

    print() {
        let stringBuilder = "|——|\n";
        for (let i = this.items.length; i > 0; i--) {
            stringBuilder +=  `| ${this.items[i-1]} |\n|——|\n`;
        }
        console.log(stringBuilder);
    }
}
```

```javascript
class Stack {

    constructor() {
        this.items = [];
        this.cursor = 0;
    }

    push(element) {
        this.items[this.cursor] = element
        this.cursor++
    }

    pop() {
        if (this._isEmpty()) {
            return "Underflow";
        }
        const poped = this.items[this.cursor]
        this.items[this.cursor] = null
        this.cursor--
        return poped
    }

    top() {
        return this.items[this.cursor];
    }

    _isEmpty() {
        return this.cursor === 0;
    }

    size() {
        return this.cursor + 1;
    }

    print() {
        let stringBuilder = "|——|\n";
        for (let i = this.size(); i > 0; i--) {
            stringBuilder +=  `| ${this.items[i-1]} |\n|——|\n`;
        }
        console.log(stringBuilder);
    }
}
```


MakeAGIF.com

```
class Queue {

  constructor() {
    this.data = []
  }

  enqueue(item) {
    this.data.push(item)
  }

  dequeue() {
    return this.data.shift()
  }

  _isEmpty() {
    return this.data.length === 0
  }

  peek() {
    return this.data[0]
  }

  size() {
    return this.data.length
  }

  print() {
    console.log(this.data)
  }

}
```

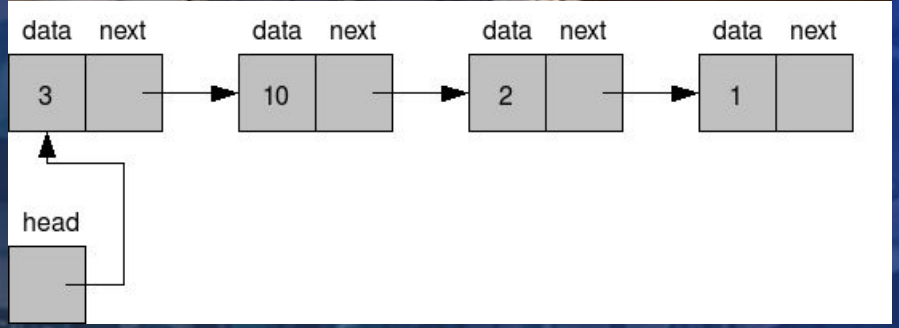
ENQUEUE

DEQUEUE

PEEK

SIZE

PRINT


data
next
3
data
next
10
data
next
2
data
next
1
head

```
class LinkedList {

  constructor() {
    this.head = null;
  }

  add(node) {
    if (this.head === null) {
      this.head = node;
    } else {
      let current = this.head
      while (current.next !== null) {
        current = current.next
      }
      current.next = node
    }
  }

  insert(index, node) {
    if(index < 0) throw new Error('Invalid index')

    if (index === 0) {
      this.add(node)
      return
    }
    let current = this.head
    let previous = null
    let i = 0
    while (i < index) {
      previous = current
      current = current.next
      i++
    }
    previous.next = node
    node.next = current
  }
```

ADD

INSERT

```
remove(index){
  if(index < 0) throw new Error('Invalid index')

  if(index === 0) {
    this.head = this.head.next
    return
  }

  let current = this.head
  let previous = null
  let i = 0
  while (i < index) {
    previous = current
    current = current.next
    i++
  }
  previous.next = current.next
}

indexOf(element){
  let current = this.head
  let index = 0
  while(current){
    if(current.value === element) return index
    index++
    current = current.next
  }
  return -1
}
```

REMOVE

INDEX OF

```
clear() {
    this.head = null
  }

  getFirst() {
    return this.head
  }

  getLast() {
    let current = this.head
    while (current.next) {
      current = current.next
    }
    return current
  }

  size() {
    let current = this.head
    let count = 0
    while (current) {
      count++
      current = current.next
    }
    return count
  }
```

CLEAR

GET FIRST

GET LAST

SIZE

```
  print() {
    let current = this.head
    while (current) {
      console.log(current.value)
      current = current.next
    }
  }

}

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

PRINT

NODE class

BLOCO #90

Timestamp

Hash Block #89

Transação 3012
Transação 3013
Transação 3014
Transação 3015
Transação 3016

Proof of work #090

Hash Block #090

BLOCO #91

Timestamp

Hash Block #90

Transação 3017
Transação 3018
Transação 3019
Transação 3020
Transação 3021

Proof of work #091

Hash Block #091

BLOCO #92

Timestamp

Hash Block #91

Transação 3022
Transação 3023
Transação 3024
Transação 3025
Transação 3026

Proof of work #092

Hash Block #092

BLOCO #93

Timestamp

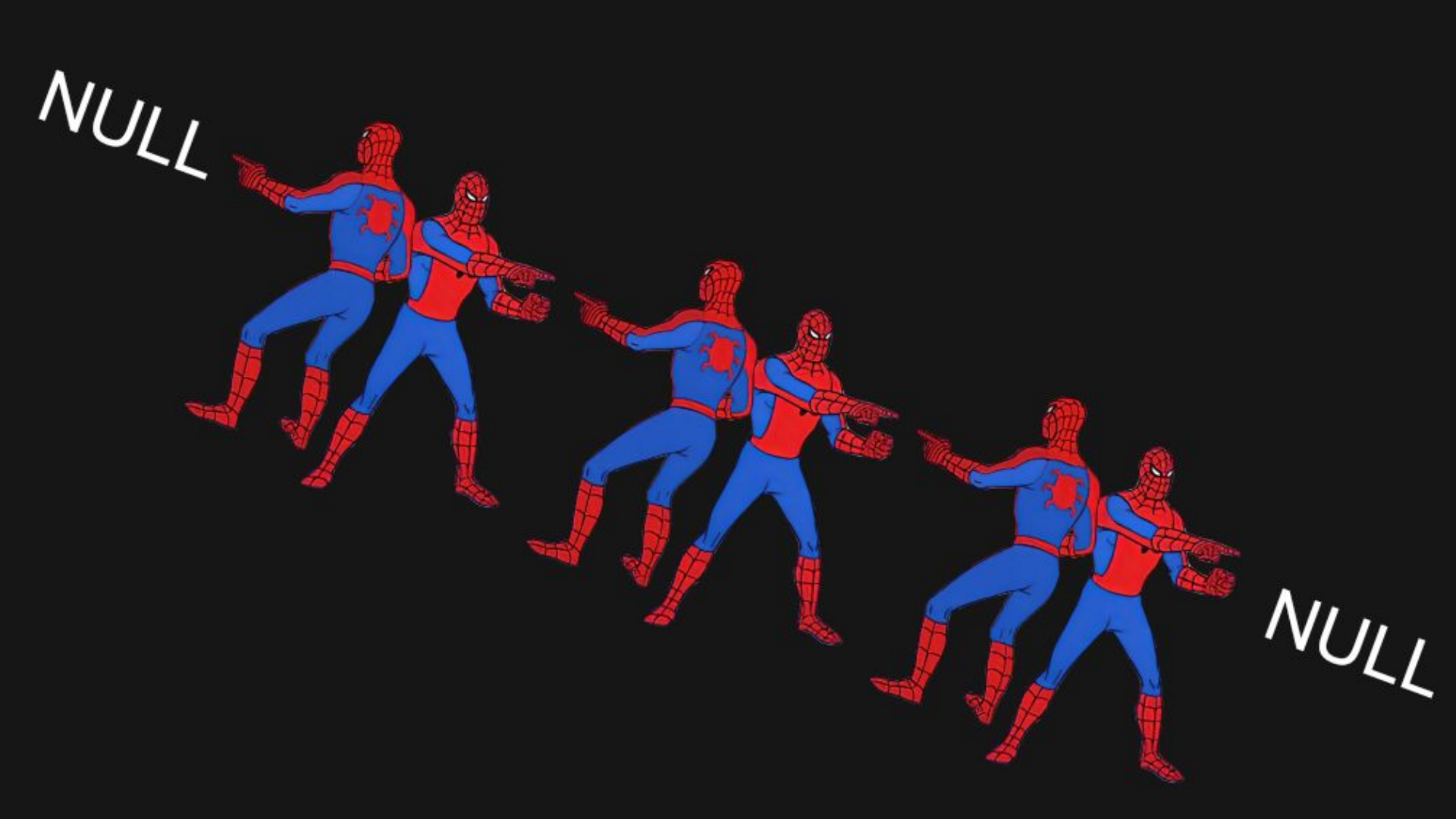
Hash Block #92

Transação 3027
Transação 3028
Transação 3029
Transação 3030
Transação 3031

Proof of work #093

Hash Block #093


NYPD


NULL
NULL

Com base na lista encadeada...

Blockchain
👌

Lista Duplamente Encadeada
👍👍

Lista Circular
👏👏👏

https://github.com/jcbombardelli/grupo-estudo-8-gama-academy