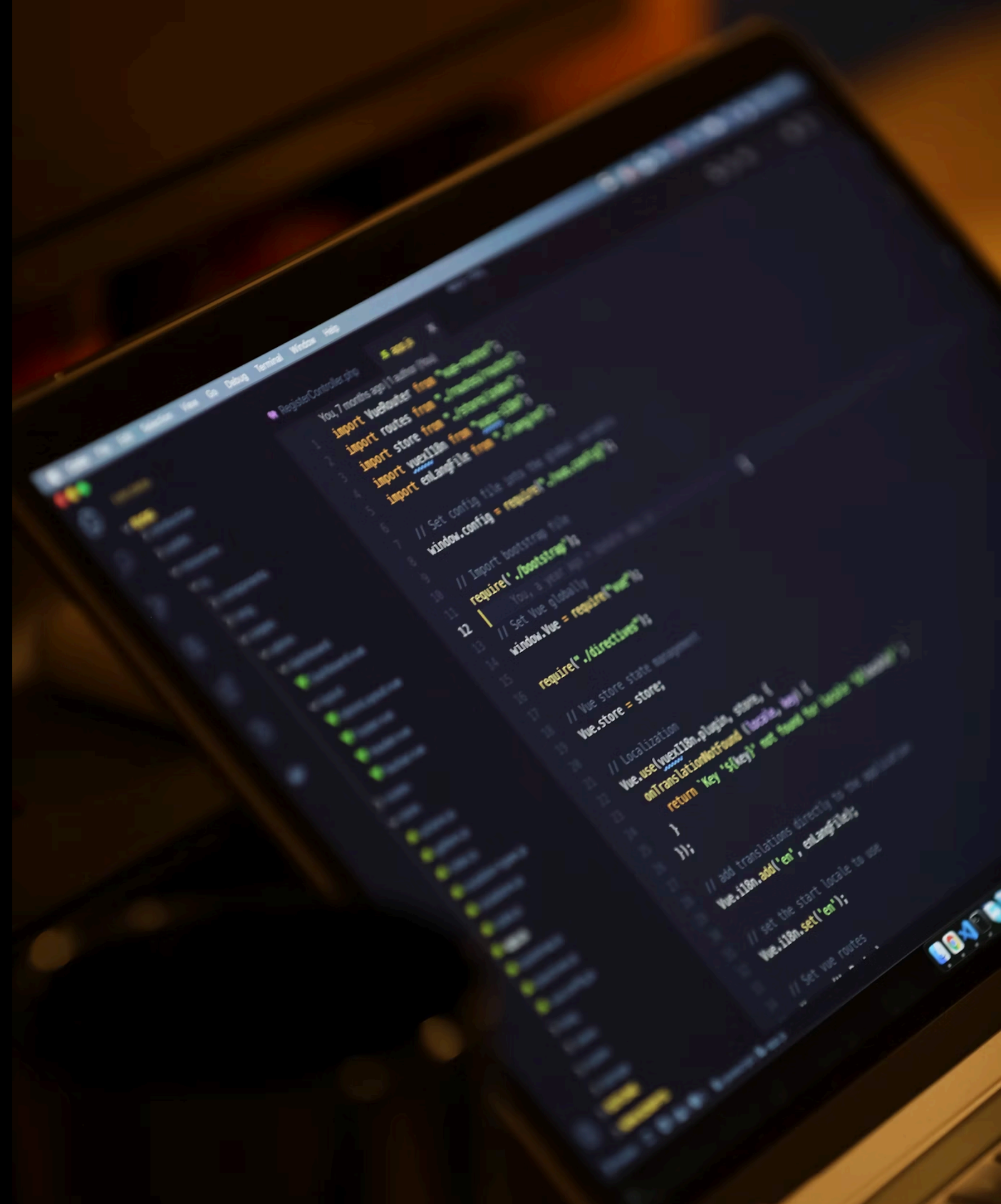
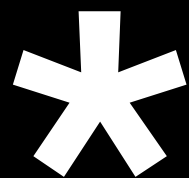


Lógica de Programação

Lucas Johann

Tópicos da aula

- funções;
- manipulação de listas;
- operadores lógicos;
- estruturas condicionais;
- exercícios



Funções JS

Uma função é um conjunto de instruções que executa uma tarefa ou calcula um valor.

Estrutura de uma função:

```
function nomeDaFunção(argumento) {  
    return retornoDesejado  
}
```

Exemplo de função

```
function soma(a, b) {  
  return a + b;  
}
```

```
console.log(soma(4, 2)) // 6
```

Função de seta (arrow function)

```
const soma = (a, b) => a + b;  
  
console.log(soma(4, 2)); // 6
```

```
const soma = (a, b) => {  
  return a + b;  
};  
  
console.log(soma(4, 2)) // 6
```

Manipulação de Listas

Quando temos uma lista, na programação chamamos de **array**, podemos adicionar itens, remover, alterar valores, contar número de itens, dentre outras manipulações.

Exemplo de lista:

```
const frutas = ["Maçã", "Abacaxi", "Laranja"]
```

Acessar index do Array

```
var primeiroItem = frutas[0]      // Maça
```

```
var tamanhoLista = frutas.length  // 3
```

```
var ultimoItem = frutas[frutas.length -1]  // Laranja
```

Adicionar um item ao final do array

```
var adicionarFim = frutas.push(Tomate)  
// ["Maçã", "Abacaxi", "Laranja", "Tomate"]
```

Remover item do início do array

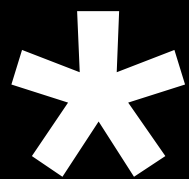
```
var removerInicio = frutas.shift()  
// ["Abacaxi", "Laranja", "Tomate"]
```


Adicionar item ao início do array

```
var adicionarInicio = frutas.unshift("Morango")  
// ["Morango", "Maçã", "Abacaxi", "Laranja", "Tomate"]
```

Procurar o índice de um item

```
var removerItem = frutas.splice(frutas, 1)  
// ["Maçã", "Abacaxi", "Laranja", "Tomate"]
```



Exercícios

arrays com JS

Operadores lógicos

operador **AND**: ambos as condições devem ser verdadeiras

Representado por **&&**

operador **OR**: pelo menos uma das condições precisa ser verdadeira,

Representado por **||**

Estruturas condicionais

A estrutura condicional tem o sentido de pergunta se, se não se e se não.

Estrutura de uma condição:

```
if (condição) {  
    retornoDesejado  
} else {  
    retornoDesejado  
}
```

Exemplo de condição

```
const idade = 16

✓ if (idade > 18) {
  |   console.log("Você é maior de idade!")
✓ } else {
  |   console.log("Você é menor de idade!")
  | }
}
```

Exemplo de condição

```
const idade = 20;

if (idade < 13) {
  console.log("Você é uma criança.");
} else if (idade ≥ 13 && idade < 20) {
  console.log("Você é um adolescente.");
} else if (idade ≥ 20 && idade < 60) {
  console.log("Você é um adulto.");
} else {
  console.log("Você é um idoso.");
}
```

Laços de Repetição

Os laços de repetição são utilizados para realizar tarefas de forma repetida baseada em uma condição. O laço se encerra somente quando a condição se tornar falsa.

São exemplos de laço de repetição:

for, for in, for of, while e do while

Laço for

Estrutura do laço:

```
for (inicialização; condição; expressãoFinal) {  
    // instruções a serem executadas repetidamente  
}
```

```
for (let i = 0; i < 9; i++) {  
    console.log(i); // Mostrará os números de 0 à 8  
}
```


Laço while

Estrutura do laço:

```
while (condição) {  
    // instruções  
}
```

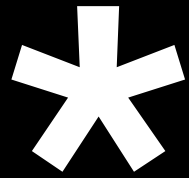
```
let i = 1;  
  
while (i < 10) {  
    console.log(i);  
    i++;  
} // Mostrará os números de 1 à 9
```

Laço do while

Estrutura do laço:

```
do {  
    // instruções  
} while (condição);
```

```
let i = 1;  
  
do {  
    console.log(i);  
    i++;  
} while (i < 10); // Mostrará os números de 1 à 9
```



Exercícios

laços de repetição com JS