

V-Web

Módulo Backend - Dia 1



JavaScript



```
function buscaBinaria(vetor, inicio, fim, k){
  if(inicio > fim) return -1;

  let mediana = Math.floor((inicio + fim)/2);

  if(vetor[mediana] == k) return mediana;
  if(vetor[mediana] < k) return buscaBinaria(vetor,
mediana, fim, k);
  if(k < vetor[mediana]) return buscaBinaria(vetor,
inicio, mediana, k);
}
```

Sintaxe JavaScript

A sintaxe do JavaScript é extremamente familiar para alguém que já usou linguagens como C, C++ ou Java.

A linguagem se diferencia em suas abstrações e açúcares sintáticos

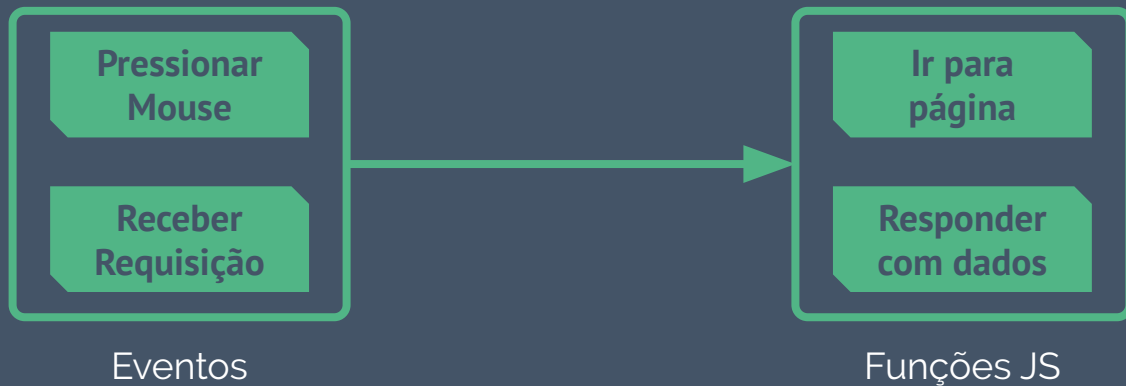


Particularidades do JavaScript

- Orientação a Eventos
- Tipagem dinâmica e fraca
- var, let & const
- Objetos em JavaScript
- Anonymous Functions & Callback Functions
- Controle de erros



Orientação a Eventos



Tipagem Dinâmica e Fraca

```
/* Tipagem dinâmica */  
let x = 0 /* x é do tipo Number */  
x = "Olá" /* x agora é do tipo String */  
  
/* Tipagem fraca */  
console.log(x + 2) /* Imprime Olá2 na tela */
```



Declaração de Variáveis: var, let e const

```
const c = 0
c = 1 /* Erro ao assinalar valor a constante */
let outer = "let externo"
if (outer == 1){
  let inner = "let interno"
  var x = "variável criada com var"
  console.log(inner)
}
console.log(inner) /* Erro: inner pertence ao if */
console.log(outer) /* outer existe, pois foi criada fora
do if */
console.log(x) /* Dá certo porque x é var */
```



Anonymous & Callback Functions

```
function(/* argumentos */){  
    /* corpo */  
}  
  
(/* argumentos */) => /* corpo */  
  
funcao_com_callback((args) => {...})
```



Objetos em JavaScript

```
let objeto = {  
  texto: "valor", /* Propriedade */  
  getText: function(){ /* Método */  
    return this.texto;  
  }  
}
```



Controle de Erros

```
try{
    /* Código com possibilidade de erro */
}catch(e){
    /* Código que cuida do erro */
}

if (comportamento_inesperado){
    throw "Nome do erro"
}
```





node.js







Arquitetura não bloqueante







let's code!





node package manager (npm)





ExpressJS

