

SEJAM BEM-VINDOS!



Jornada DEV AM. CE. RR

Formação gratuita
em Desenvolvimento
Front-End SENAI



Começamos em breve!

A aula vai iniciar em **5 minutos**.

Relaxe e aproveite a música enquanto aguardamos todos chegarem!



Lógica com JavaScript #11

Missão de hoje

- O que são classes
- Porque precisamos de classes?
- Conceito de instanciação
- Criar suas próprias classes
- Método construtor
- Variáveis de instância
- Métodos de instância

O que é uma classe no JavaScript?

Uma classe é uma forma especial de escrever funções construtoras e prototipagem de maneira mais clara e organizada.

Ela define um **modelo** (molde) para criar **objetos** com **propriedades** e **métodos**.

Em vez de criarmos funções e adicionarmos métodos manualmente, a classe já agrupa tudo de forma mais intuitiva.

Por que precisamos de classes?

As classes permitem **agrupar variáveis e funções** que descrevem uma entidade (usuário, produto, receita ...).

Isso torna o código mais **organizado, reutilizável e fácil de manter.**

Diferença importante:

Objetos: agrupam apenas variáveis (pares chave-valor)

Classes: agrupam variáveis (propriedades) + funções (métodos)



Uma classe é a planta de uma casa;

cada instância é uma casa construída.



Instanciação: Criando objetos

Instanciar significa criar um objeto a partir de uma classe usando a palavra reservada `new`.

"A classe é a fábrica e as instâncias são os produtos fabricados.

Cada instância é um **objeto independente** com suas próprias propriedades e métodos.

```
● ● ●  
class Pessoa {  
  constructor(nome) {  
    this.nome = nome;  
  }  
}  
  
const pessoa1 = new Pessoa("Sueli Machado");  
const pessoa2 = new Pessoa("Carlos Almeida");  
console.log(pessoa1 === pessoa2); // false
```

Classe Vs Instância

Classe:

É o modelo, a "receita" para criar objetos.

Instância:

É o objeto concreto criado a partir da classe.

Importante:

Cada instância é unida e diferente de qualquer outra, mesmo que criada com os mesmos parâmetros.

```
class Pessoa {  
  constructor(nome) {  
    this.nome = nome;  
  }  
  
const pessoa1 = new Pessoa("Sueli Machado");  
const pessoa2 = new Pessoa("Carlos Almeida");  
console.log(pessoa1 === pessoa2); // false
```

Definindo sua própria classe

Para criar uma classe em Javascript,
use:

1. **Palavra-chave:** `class`
2. **Nome:** deve seguir o padrão
UpperCamelCase
3. Chaves `{}` para delimitar o corpo

```
class Usuario {  
}  
  
class Produto {  
}
```

Método construtor de classe

O **constructor** é um método especial:

- Método especial chamado **automaticamente** ao criar uma instância
- Usado para **inicializar** o objeto
- **Não** usa a palavra-chave **function**

```
● ● ●  
class Usuario {  
  constructor() {  
    console.log("Criando uma instância...");  
  }  
}  
  
const u1 = new Usuario();  
// → Criando uma instância...
```

Intervalo!

Finalizamos o nosso primeiro período de hoje.

Nos vemos em 15min.

CE

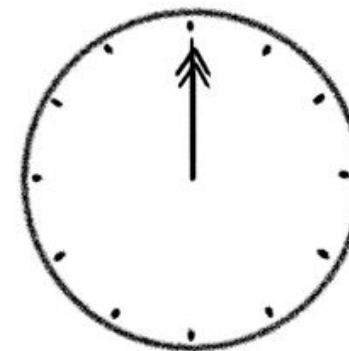
Início: 20:00

Retorno: 20:15

AM. RR

Início: 19:00

Retorno: 19:15



Construtor de classe com parâmetros

Assim como funções, o construtor pode receber **parâmetros** para inicializar o objeto com específicos valores.

Os parâmetros são passados ao criar a instância com `new`.

```
class Usuario {  
    constructor(nome, sobrenome) {  
        console.log("Criando usuário:");  
        console.log(nome, sobrenome);  
    }  
  
    const usuario1 = new Usuario("Sérgio", "Diniz");  
    const usuario2 = new Usuario("Carla", "Alves");
```

Criando variáveis de instância

Variáveis de instância são criadas dentro do construtor usando o **this**.

this se refere á instância atual da classe.

Benefícios:

- As variáveis ficam **armazenadas no objeto**
- Pode ser **acessadas por todos os métodos** da classe
- Cada instância seus **próprios valores**.



```
class Usuario {  
  constructor(nome, sobrenome) {  
    this.nome = nome;  
    this.sobrenome = sobrenome;  
  }  
}  
  
const usuario = new Usuario("Luiz", "Moura");  
console.log(usuario.nome); // "Luiz"
```

Atividades na plataforma

<LAB365>

