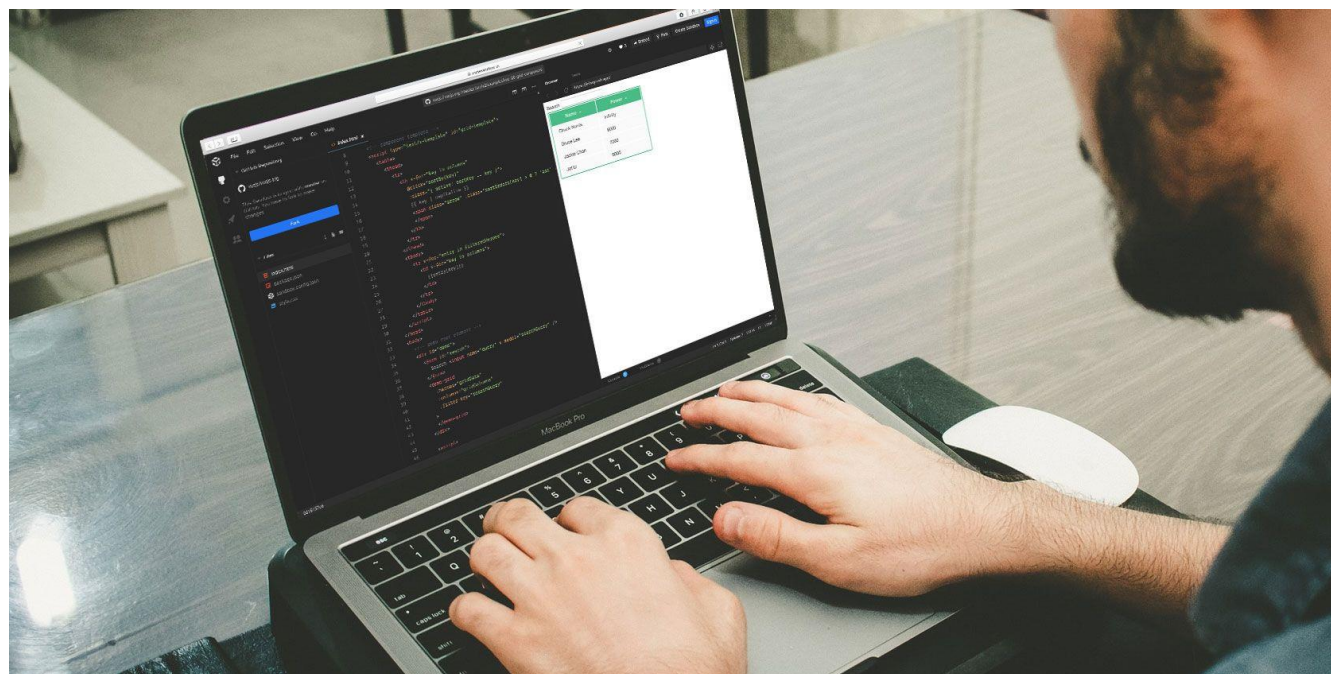




# Desenvolvimento Front-End



Professora Ma. Tatiana Aparecida de Almeida

## Aula 2: Introdução ao Vue.js

### Sumário:

- 1.Introdução ao Vue.js
- 2.O que são componentes Vue.js
- 3.Conceitos-chave:
  1. Interpolação
  2. Diretivas (v-if, v-for, v-bind, v-model)
  3. Eventos
  4. Computed Properties
- 4.Códigos de Exemplo
- 5.Exercícios práticos

# 1.Introdução ao Vue.js

Vue.js é um **framework JavaScript progressivo** que ajuda a criar **interfaces de usuário** de forma eficiente.

O termo "progressivo" significa que você pode começar com pequenas partes e adicionar funcionalidades conforme necessário.

Ele é flexível e pode ser usado tanto para projetos simples quanto para **Single-Page Applications (SPAs)** complexas.

## 2. O que são componentes Vue.js

Os componentes são blocos reutilizáveis que encapsulam HTML, JavaScript e CSS. No Vue.js, eles são o coração da estrutura de desenvolvimento.

vue

Copiar código

```
<template>
  <div>
    <h1>{{ mensagem }}</h1>
  </div>
</template>

<script>
export default {
  data() {
    return {
      mensagem: "Olá, Vue.js!"
    };
  }
};
</script>


<style scoped>
h1 {
  color: blue;
}
</style>
```

## 3. Conceitos-chave:

### 3.1 Interpolação:

A interpolação é usada para exibir dados dinâmicos em uma página HTML.

vue

 Copiar código

```
<template>
  <p>{{ mensagem }}</p>
</template>


<script>
export default {
  data() {
    return {
      mensagem: "Bem-vindo ao Vue.js!"
    };
  }
};
</script>
```

## 3.2 Diretivas Vue:

### `v-if` - Condicional:

Controla se um elemento deve ser renderizado com base em uma condição.

vue

 Copiar código


```
<template>
  <p v-if="mostrarTexto">Este texto é exibido quando a condição for verdadeira.</p>
</template>

<script>
export default {
  data() {
    return { mostrarTexto: true };
  }
};
</script>
```

## v-for - Iteração:

Usado para renderizar uma lista de itens.

vue

 Copiar código


```
<template>
  <ul>
    <li v-for="(item, index) in itens" :key="index">{{ item }}</li>
  </ul>
</template>

<script>
export default {
  data() {
    return { itens: ["Item 1", "Item 2", "Item 3"] };
  }
};
</script>
```

## v-model - Ligação de dados bidirecional:

Permite vincular o valor de um campo de entrada a uma variável.

vue

 Copiar código

```
<template>
  <input v-model="nome" placeholder="Digite seu nome" />
  <p>Seu nome: {{ nome }}</p>
</template>


<script>
export default {
  data() {
    return { nome: "" };
  }
};
</script>
```



## v-bind - Ligação de atributos:

Vincula atributos HTML a uma variável.

vue

 Copiar código


```
<template>
  
</template>

<script>
export default {
  data() {
    return { imageUrl: "https://exemplo.com/imagem.png" };
  }
};
</script>
```

### 3.3 Eventos Vue.js:

Eventos são interações que podem disparar funções. O `@click` é uma das diretivas mais comuns.

vue

 Copiar código


```
<template>
  <button @click="saudar">Clique aqui</button>
</template>

<script>
export default {
  methods: {
    saudar() {
      alert("Olá, Vue.js!");
    }
  }
};
</script>
```

### 3.4 Computed Properties:

São propriedades que dependem de outras variáveis e são recalculadas automaticamente quando essas variáveis mudam.

vue


 Copiar código

```
<template>
  <p>Mensagem invertida: {{ mensagemInvertida }}</p>
</template>

<script>
export default {
  data() {
    return { mensagem: "Vue.js" };
  },
  computed: {
    mensagemInvertida() {
      return this.mensagem.split('').reverse().join('');
    }
  }
};
</script>
```

## Exemplo 1: Contador Simples

vue

 Copiar código

```
<template>
  <div>
    <p>Contador: {{ contador }}</p>
    <button @click="incrementarContador">Incrementar</button>
  </div>
</template>
```

```
<script>
export default {
  data() {
    return { contador: 0 };
  },
  methods: {
    incrementarContador() {
      this.contador++;
    }
  }
};
</script>
```

### Seção `<template>`:

1. `<template>`:
  - Define o bloco HTML do componente Vue. Tudo que está dentro dessa tag será exibido na página.
2. `<div>`:
  - Um contêiner geral que agrupa o conteúdo do componente.
3. `<p>Contador: {{ contador }}</p>`:
  - Exibe o valor da propriedade `contador` usando a interpolação `{{ }}`. O valor `contador` é gerenciado no objeto `data` do componente.
4. `<button @click="incrementar">Incrementar</button>`:
  - Cria um botão que dispara um evento de clique (`@click`) ao ser pressionado. Esse evento executa o método `incrementar`.

## 5. Exercícios Práticos

### Exercício 1: Contador de Cliques

**Objetivo:** Crie um contador que diminui e aumenta com dois botões.

- Um botão incrementa o contador.
- Outro botão decrementa o contador.

### Exercício 2: Mostrar/Ocultar Texto

**Objetivo:** Crie um botão que ao ser clicado mostre ou oculte um texto na tela usando v-if.

### Exercício 3: Lista Dinâmica

**Objetivo:** Crie uma lista onde o usuário pode adicionar itens e também remover itens clicando neles.

### Exercício 4: Input de Nome

**Objetivo:** Crie um campo de entrada onde o usuário pode digitar seu nome e, ao lado, exiba "Bem-vindo, [nome digitado]!" em tempo real.

## Referência de Estudo

<https://blog.tiagomatos.com/vue-js-diretivas-v-show-v-if-v-else/>